Human localization and activity classification by machine learning on Wi-Fi channel state information

Ruslan Lagashkin

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 17.08.2020

Supervisor

Prof. Simo Särkkä

Advisor

Dr Mikko Honkala

Aalto University
School of Electrical
Engineering

Aalto University
School of Electrical Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Ruslan Lagashkin

Title Human localization and activity classification by machine learning on Wi-Fi channel state information

Degree programme Electronics and Nanotechnology

Major Micro- and Nanosciences **Code of major** ELEC3037

Supervisor Prof. Simo Särkkä

Advisor Dr Mikko Honkala

Date 17.08.2020 **Number of pages** 81+3 **Language** English

Abstract

Devices communicating via Wi-Fi adjust subcarrier correction coefficients in real time. The stream of correction coefficients for all subcarriers is called channel state information (CSI). The latter can be used for human body sensing, in particular current activity and location. The thesis aims to create a robust, environment-agnostic activity classifier. In other words, a neural network (NN) trained to recognize and classify human actions in one location should not dramatically lose prediction capability if transferred to another. This purpose has been achieved in three steps. First, for a neural network to abstract from a particular environment, a diverse data has to be collected. Therefore, a dedicated laboratory equipment to automate a Wi-Fi access point (AP) physical movement and rotation has been developed and constructed. After data for training NNs has been collected, the environment-specific information has been cleaned by classic signal processing algorithms. Finally, a dedicated neural network architecture adjustments have been implemented. Altogether, the goal of environment agnostic classification for the target "sit down", "stand up", "lie down", and "unlie" activities is achieved. However, classification accuracy depends on the similarity between train and test human subjects.

The work argues that activity classification and localization tasks have orthogonal goals and focus on different aspects of CSI information. In particular, activity classification NN is interested in ongoing physical movement features and should work independently of the human location. On the contrary, localization NN should ignore subject activities and infer only positioning. Therefore, these two tasks are separated into standalone NN architectures. Since environments such as apartments can be very different from each other, it is assumed that training a universal NN localizer is not possible. Since the localization NN needs to be re-trained for each particular environment, its virtue would be an inexpensive training cycle. Tho achieve this goal, localization NN is substantially reduced in size from 58.8 to 0.4 million parameters.

Keywords Wi-Fi, channel state information, activity classification, multi-environment, invariant, room, localization, data augmentation, neural network, convolutional neural network, architecture, machine learning

Preface

Hereby I want to express the gratitude to Leo Kärkkäinen for connecting people and making this thesis possible. By the virtue of his effort, I have been unjustly fortunate to work under Swetha Muniraju's direction who has offered the luxury of creative freedom and encouraged area research. I want to thank Mikko Honkala for prompt and professional advice whenever it was required throughout the process. I want to thank Simo Särkkä for an exceptional patience while guiding and highlighting the untold pitfalls of academic writing. I am grateful to Akos Vetek and István Beszteri who helped to carry on the heavy part of the project.

But most of all, I want to express the deepest gratitude to my parents for constant moral support during the bright and dim moments of this venture.

Otaniemi, 20.05.2020

Ruslan Lagashkin

Contents

Symbols and abbreviations

Abbreviations

AP	Access point
CNN	Convolutional neural network
CSI	Channel state information
DC	Direct current
DMA	Direct memory access
EM	Electromagnetic
GPIO	General-purpose input/output pins
LSTM	Long short-term memory
MIMO	Multiple input, multiple output
ML	Machine learning
NN	Neural network
PEAD	Physical environment data augmentation device
PVC	Polyvinyl chloride
PWM	Pulse-width modulation
RAM	Random access memory
RGB	Red, green, and blue additive color model
RF	Radio frequency
RPi	Raspberry Pi
RSS	Received signal strength
SGD	Stochastic gradient descent
STA	Station device of Wi-Fi network

1 Introduction

Originally, Wi-Fi was developed solely for data transmission [O'Sullivan et al., 1993]. However, existing Wi-Fi infrastructure may also be applied to human body sensing [Liu et al., 2014] based on the principle described below.

Wirelessly communicating devices use the surrounding environment as a transmission channel. A human body with unmatched impedance may enter or move within the channel, causing input impedance variation. Wi-Fi device compensates for the variation by applying a correction coefficient for each subcarrier [Tulino et al., 2005]. The correction coefficient contains amplitude and phase adjustment. An array of such complex correction coefficients for all subcarriers is called channel state information (CSI). Channel state information is updated in real time and imprints ongoing human movements (see Figure 1). It is then forwarded to neural networks (NNs) which try to reconstruct what is going on the signal's way.

Within the scope of the present work, NNs are used for two purposes on Wi-Fi CSI. First, to classify human physical actions or "activities" such as *sitting down* on a chair, *standing up*, *lying down* on a floor or bed, and *getting up*. Second, to localize one human within an apartment, in other words, to predict which room is the test subject located in.

Regarding the activity classification purpose, a trained network could produce good predictions only when both Wi-Fi access point (AP) and station (STA) remained completely stationary. If one of the devices was shifted by just one meter, classification accuracy would degrade dramatically. Consequently, this work aims to build a robust, "environment-agnostic" classifier, tolerant to Wi-Fi devices displacements and capable of operating in new, previously unseen environments.

First of all, for a neural network to abstract from environment-specific CSI features, it requires data from several places. To achieve this goal, two data collection spaces in the USA and China have been supplemented with other two distinct Finnish environments. Subsequent work raises the following research questions along the whole machine learning pipeline:

1. How to generate more diverse data from a still limited number of physical CSI collection spaces?

2. Could supplying the CSI phase aside amplitude be beneficial for NN inference?

3. Which data augmentation methods clear the signal waveform from environment-specific information, while leaving the imprinted activity signatures intact?

4. Is it possible to modify and regularize an existing neural network architecture to improve the cross-environment prediction accuracy?

In order to address the stated questions, the following actions have been taken. (1) It has been observed that displacing a Wi-Fi device with respect to the surrounding environment changes subcarriers' multipath configuration, as can be seen in Figure 9. This, effectively, creates a novel viewpoint of an old environment and, arguably, as good as recording CSI in a new place. However, manually moving CSI

acquisition devices is routine and laborious. Therefore, simple robots to automate this procedure have been constructed. These machines were distributed to all data collection facilities. After data recording, (2) it has been found that the CSI phase is preferable compared to the amplitude for environment agnostic activity classification. Furthermore, (3) useless and useful data augmentation and processing methods have been identified. Finally, (4) the standard computer vision convolutional architecture has been specifically adjusted for CSI data. Certain adjustments allowed to deploy a set of custom *between-parallel-branches* regularizing losses that contributed to the resulting prediction accuracy.

Regarding the body localization purpose, it appears that apartments can be very different from one another. Therefore, a neural network needs to be re-trained for each individual environment. Consequently, this work aims to shorten the training time of a localization NN by making it smaller.

In the end, the thesis demonstrates substantially leaner localization neural network architecture as well as environment agnostic activity classification. Furthermore, it outlines the possible directions for future development.

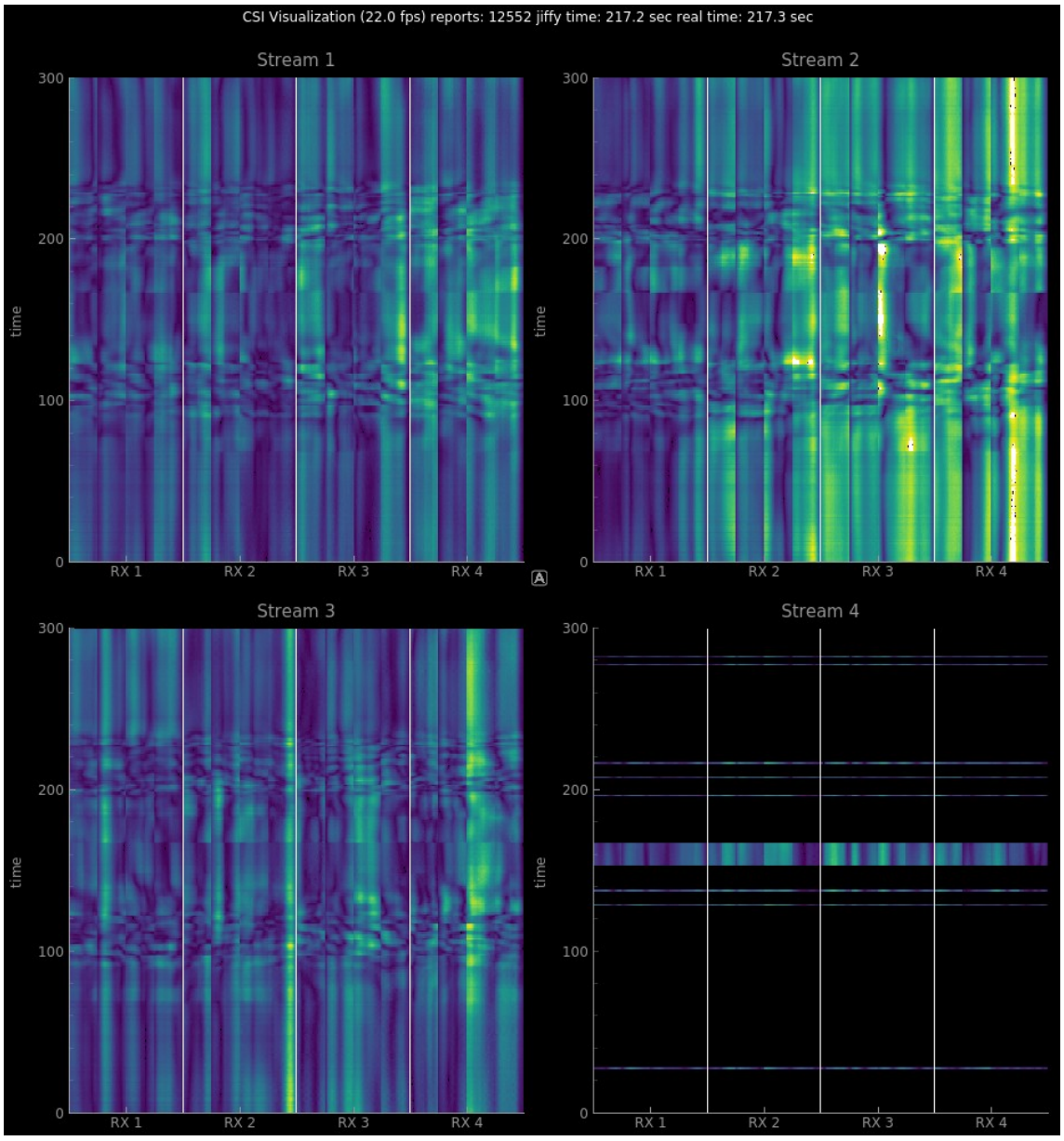


Figure 1: Amplitude of the channel state information stream for $(4_{AP} \times 4_{STA})$ antennas. The horizontal axis is for subcarriers and the vertical is for time. The stream is interpolated to 100 Hz and hence 300 readings yield the total visualization depth of 3 seconds. Within the time frame ~0.9 to ~2.4 seconds, an activity of waving a hand back and forth in front of the STA takes place. Brighter colors represent a higher subcarrier correction coefficient amplitude at a given moment.

2 Background

This chapter begins by identifying wireless sensing as an artificial method of human perception with novel properties in Section 2.1. Section 2.2 collects relevant research in the topic and determines the primary approach. Section 2.3 proceeds with the description of the channel state information collection equipment. Section 2.4 presents prior art of methods used for NN architecture modification. Finally, Section 2.5 assesses prior art of rope knots for *physical environment augmentation device* construction.

2.1 Sensing via Wi-Fi radiation

For a long time, primary instruments to detect presence, localize and classify actions of a fellow being were eyes and ears. These instruments rely on photons and phonons respectively.

Human eye can typically perceive photons with wavelength 380 to 740 nm [Starr, 2005] and its diffraction-limited resolution is less than 0.5 um. The disadvantage of such a short wavelength is that objects behind even small opaque macroscopic obstacles are practically invisible.

Human ear can typically hear sounds in the 20 Hz - 20 kHz range [Rossing, 2014]. However, after approximately 14 kHz, threshold for perceiving sound rises abruptly – in other words, only loud noises are heard [Ashihara, 2007]. This leaves humans with a well-detectable phonon wavelength range between ∼17 m to ∼25 cm. Although such wavelengths are bending around macroscopic objects, they do not provide sufficient resolution for smaller than ∼10 cm objects.

In comparison, Wi-Fi wavelengths are 12.5 cm and 6 cm for 2.5 GHz and 5 GHz respectively. This is shorter than the ∼25 cm of human-detectable phonons limit and is sufficient for recognizing, for example, individual body parts [Adib et al., 2015].

Taking into account over two decades of technology optimization [O'Sullivan et al., 1993] as well as the present widespread of Wi-Fi infrastructure, sensing objects with this type of communication network may be considered as a viable addition to other means of perception. Since the frequencies of S and C bands[1] are undetectable with natural human senses at low power, an introduction of consumer-level Wi-Fi human body localization and activity classification product rises certain ethical concerns that are described in Section 5.2.

2.2 Research in Wi-Fi CSI human sensing sphere

Twelve years after Wi-Fi technology first introduction for home use [Khalili et al., 2020], a diagnostic tool for retrieving real-rime CSI information via the famous *Intel 5300 NIC* was released [Halperin et al., 2011]. Two years later, Yang et al. [2013] pointed out the advantage of CSI for human localization compared to the earlier used received signal strength (RSS) [Bahl and Padmanabhan, 2000]. Indeed, RSS

[1] In accordance to IEEE Std 521-2002 Standard [Belov et al., 2012]

is just a single number, while a CSI frame[2] with 104 subcarriers, 4 Rx and 4 Tx antennas contains (104, 4, 4) complex numbers with real and imaginary parts (an example of such CSI frame can be seen in Appendix A). This is 3328 times more raw information per unit time. In the following year Liu et al. [2014], while focusing on breath rate detection and sleeping pose estimation, pioneered a substantial part of modern CSI data augmentation and machine learning methods. In particular,

- outlier points removal,
- wavelet filtering,
- sample interpolation,
- feature engineering,
- principal component analysis, and
- convolutional neural network application.

Present-day Wi-Fi-based sensing technologies studied at that time include:

- Adib and Katabi [2013] estimated angle of arrival with a dedicated phased array for human counting and coarse gesture recognition in a neighboring room through a wall.
- Pu et al. [2013] used Doppler shift for house-wide gesture detection. Although ~17 Hz frequency shift has been inferior compared to 5 GHz carrier frequency and 20 MHz bandwidth, authors found a way to perform shift detection from the whole OFDM-modulated Wi-Fi signal spectrum.

Nowadays, there exists an abundance of research concerning Wi-Fi human sensing. For instance, a relatively recent survey [Ma et al., 2019] analyzes 131 papers, including 45 for activity/gesture recognition and 24 for human body localization/tracking. Three examples illustrate CSI sensitivity:

1. Ali et al. [2017] recognize keystrokes on the mechanical laptop keyboard in a continuously typed sentence with an accuracy of 93.5%, while single keys pressing with 96.4%.
2. Zheng et al. [2017] develop no-smoking alert by recognizing a specific smoking gesture and it's cyclic behavior.
3. Zhang et al. [2018a] achieve 98% accuracy in breathing rate estimation, while Wang et al. [2017a] and Wang et al. [2017b] demonstrate multi-user breathing sensing.

[2]During this frame collection, an acquisition device has been set to operate by the 802.11ac protocol. At this setting, it produces 26 subcarriers for each 20 MHz of bandwidth, which yields 104 subcarriers at the highest option of 80 MHz.

Although many solutions show good results in sterile laboratory environment, the challenge is to build a universal appliance capable of operation in varying environments. This dictates the preference for machine learning-based algorithms compared to modeling-based ones due to their inherit ability to work with "fuzzy" data [Ma et al., 2019, Fan et al., 2019]. In this area, Guo et al. [2019] compares classification of 16 activities by several machine learning methods and finds convolutional neural networks (CNN) and Long short-term memory (LSTM) to perform slightly better (above 90% overall accuracy) than other methods. The present thesis explores the CNN method.

2.3 Wi-Fi CSI acquisition devices

CSI collection for the present work has been conducted with four-antenna IEEE 802.11ac devices operating from 4.9 to 5.85 GHz and supporting 20/40/80 MHz bandwidth. These devices can function as either Wi-Fi access point (AP) or station (STA). AP supports real-time channel state information acquisition from up to three connected STAs. All CSI data in the present thesis has been collected with all the three links active.

Every 20 MHz of transmission bandwidth adds 26 subcarriers, so the maximum at 80 MHz is 104. In principle, more subcarriers yield more training information for a neural network. However, as it could be seen from Figure 1, subcarriers tend to form repeating, "similar to an interference" pattern which might limit the usefulness of extra subcarriers. Still, there is no practical reason to collect raw data with bandwidth below the upper limit of 80MHz.

The concept of a spatial stream is illustrated in the Figure 2. There, different amplitudes for the same frequency are formed at spatially-distant locations. Therefore, the number of possible spatial streams is capped by either the number of Tx or Rx antennas. The latter is equal to 4 for both devices in a link. Therefore, each link provides (4x4) multiple input multiple output (MIMO) correction coefficients matrix and allows up to 4 spatial streams. However, sometimes not all spatial streams are used. For instance, in Figure 1, capturing a typical recording situation, "Stream 4" is "flickering" and active for only short periods of time. Such streams with an insufficient number of CSI reports are filtered out and not utilized for neural network training.

Antennas of the CSI acquisition devices can be seen in Figure 18 and schematically in Figure 31. The radiation pattern of a dipole antenna can be viewed as a superposition of Hertzian dipoles radiation patterns. The electric field component of EM wave radiated by an elementary dipole has a *sin* relation to the angle from dipole orientation [Cheng, 2014]. In other words, the electric field strength in the direction perpendicular to the dipole is maximum, while in the direction along the dipole it equals to zero. Therefore, one of the effective methods to change subcarriers multipath configuration and create a novel CSI viewpoint for an existing environment is to rotate all STAs or an AP itself by 90°, which is confirmed in Figure 9.

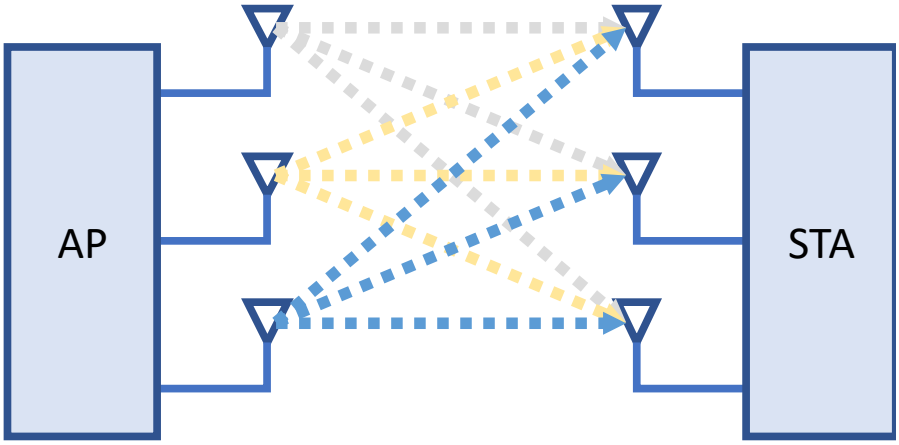
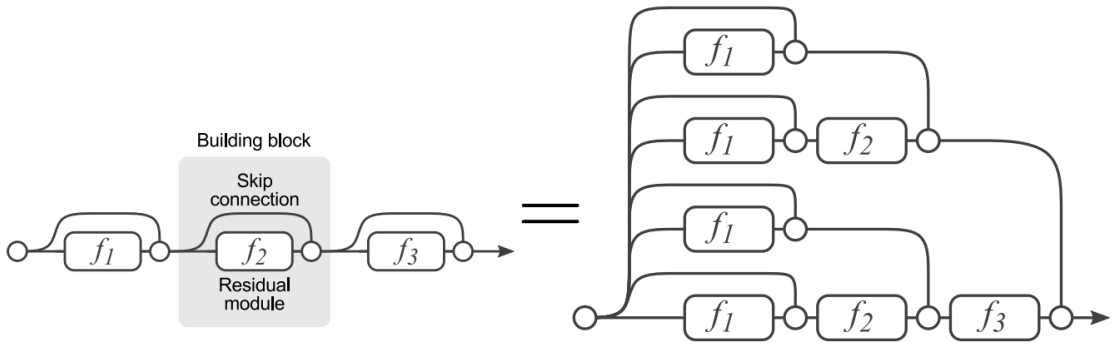


Figure 2: Illustration of a 3x3 multiple input, multiple output (MIMO) with a capability of forming maximum 3 spatial streams. If the number of STA antennas is N, and the number of AP antennas is M, the number of possible spatial streams would be $\min(N, M)$.



(a) Conventional 3-block residual network (b) Unraveled view of (a)

Figure 3: Reprinted from Veit et al. [2016]. Due to skip connections, ResNet behaves as an ensemble of networks with varying depth. For instance, dropping a layer from a branching ResNet degrades its performance only marginally, while excluding a layer from a strictly sequential VGG [Simonyan and Zisserman, 2015] drops performance dramatically Veit et al. [2016].

2.4 Machine learning methods

2.4.1 Deep feedforward neural network architectural features

Often the natural architecture choice for time series data is a recurrent neural network based on time-proven LSTM units [Hochreiter and Shmidhuber, 1997] or significantly faster [Chung et al., 2014], but weaker [Weiss et al., 2018] gated recurrent units [Cho et al., 2014]. The present work, however, applies and investigates convolutional feedforward architectures which could be used for time series data processing provided that time is unfolded in an extra dimension. For example, *1D* data with unfolded time dimension becomes *2D* and the time sequence analysis starts to resemble the well-studied [Andreopoulos and Tsotsos, 2013] computer vision tasks.

Simplest CNNs [LeCun et al., 1989] contain several sequential convolutional layers from the bottom and few fully-connected layers at the top. Convolutional layers are often interleaved with pooling layers to increase neuron's effective receptive field as well as to decrease input size for costly fully-connected layer.

In addition to the "vanilla" CNN architecture, ResNet [He et al., 2016], Inception Net [Szegedy et al., 2015] and DenseNet [Huang et al., 2017] architecture principles have survived the testing and are incorporated into the final versions of localization and activity recognition networks and hence are described here.

ResNet [He et al., 2016] introduces a so-called residual connection – in essence, a skip or "highway" connection across one or several stacked convolutional layers (see Figure 3). An output tensor from an intermediate layer is saved and later added to the output of some further layer. Note that unlike the DenseNet approach, intermediate outputs are not concatenated but element-wise summed.

DenseNet can be viewed as another approach to bypass the vanishing gradient problem as well as combine information from various sized receptive fields [Huang et al., 2017]. An illustration of a DenseNet building block can be seen in Figure 4.

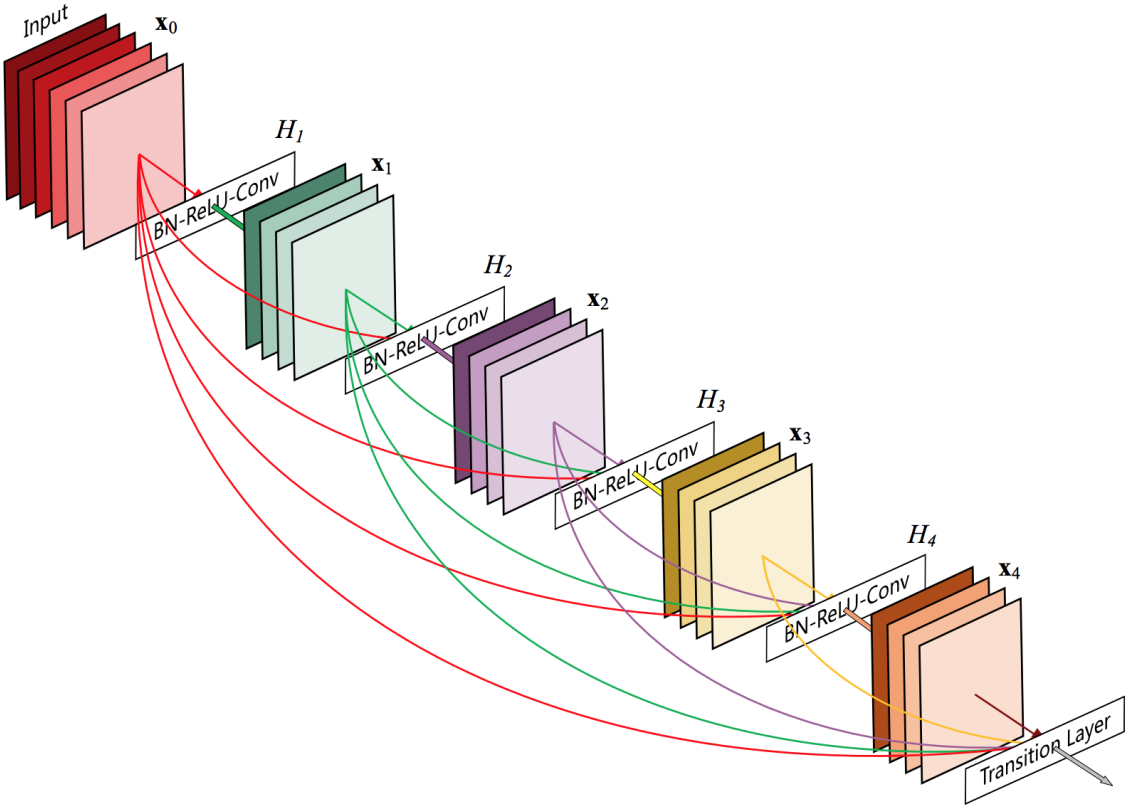


Figure 4: Reprinted from Huang et al. [2017]. DenseNet block or *dense block*. Later sub-blocks ($x_1 - x_4$) in the D*dense block* receive outputs from all previous sub-blocks. Outputs are depth-concatenated. In order to keep the depth manageable, each sub-block ends with a 1x1 convolutional layer. Therefore, outputs are squeezed with pointwise convolution prior to concatenation. Note that all activation maps within a DenseNet block should have the same dimensionality (except the dimension they are concatenated along). *Dense block* blocks are separated by *transition blocks*, which include a pooling layer.

Note that, unlike in ResNet, (1) multiple activation maps from preceding sub-blocks are used and (2) they are depth-concatenated instead of element-wise summed.

Compared to the fully-sequential network, DenseNet contains comparatively less parameters due to reuse of previous layers outputs. It, however, consumes disproportionately large amount of RAM due to the need to store intermediate results to concatenate within DenseNet blocks.

One of the main features of the first **Inception** model [Szegedy et al., 2015] is adding parallel branches with differently sized kernels with the purpose of (1) creating shallower paths to mitigate vanishing gradients problem and (2) allow grasping differently-sized patterns at a given network depth. In the context of this work, the network with unequal kernels within parallel branches would be called having an "Inception feature".

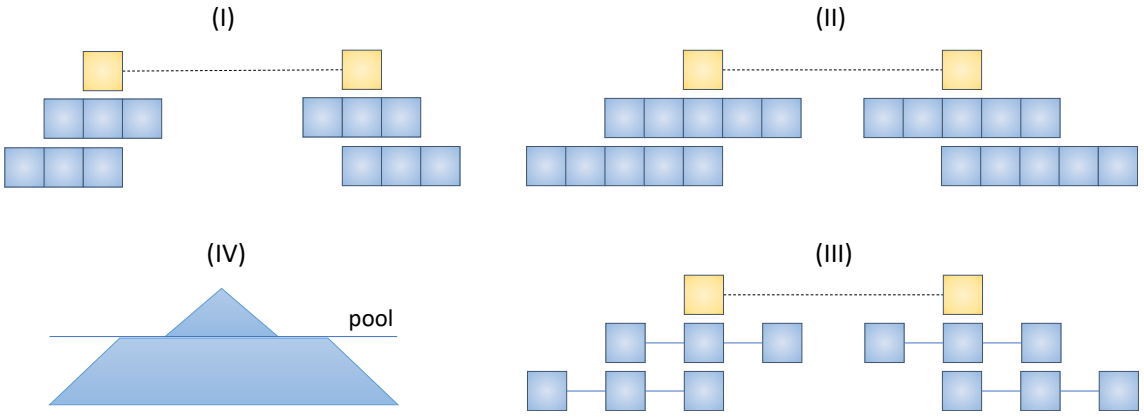


Figure 5: Effective receptive field (ERF) growth of a convolutional layer neuron. (I) Sequential 3x convolutional filters increase ERF by 1 from each side for a layer. (II) Sequential 5x convolutional filters increase ERF by 4 pixels in total per layer. (III) Dilation of >1 causes a more rapid increase of ERF for x3 filters but leaves non-perceived "gaps". (IV) Pooling increases the effective receptive field proportionally to the pooling kernel size (provided that stride is equal to the kernel size as practiced in the present work).

2.4.2 Effective receptive field of a CNN neuron

Unlike recurrent LSTM networks [Hochreiter and Shmidhuber, 1997] that compress and store the fading information about past events, CNNs [Fukushima, 1980, LeCun et al., 1999] have no such memory. Instead, they process pieces of CSI recordings of limited duration (for example, 5 seconds) independently. Therefore, (1) it is important for an entire activity or at least its distinctive feature to be within the input sample and (2) effective receptive field [Luo et al., 2016] of the top convolutional layer neuron should be sufficient to contain the whole distinctive feature. Figure 5 depicts the effect of different NN layers on the effective receptive field.

2.5 Physical environment augmentation devices

2.5.1 Reason for PEAD construction

CSI correction coefficients maintain subcarrier-specific constant offsets if AP and STA are stationary with respect to each other and the environment. However, if AP or STA is moved or rotated (see Figure 9), multipath fading and subcarriers offsets change. While a NN used to a particular configuration or a set of configurations has been exposed to an unknown subcarriers offsets distribution due to STA movement, NN has been losing an ability to perform correct activities classification.

One of the approaches to prepare an environment-agnostic NN from the data perspective, is to provide NN with diverse data from different environments. Provided the limited number of laboratory rooms dedicated for CSI data collection, these environments could at least be "augmented" by moving Wi-Fi devices within. However, frequent manual movement of devices promised to be a laborious task. In order

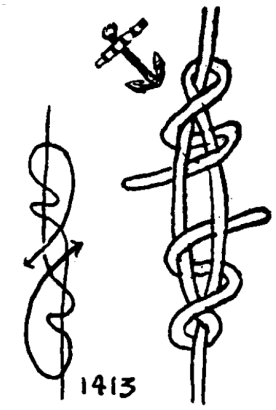


Figure 6: (1413) The blood knot. Reprinted from Ashley [1944, page 259].

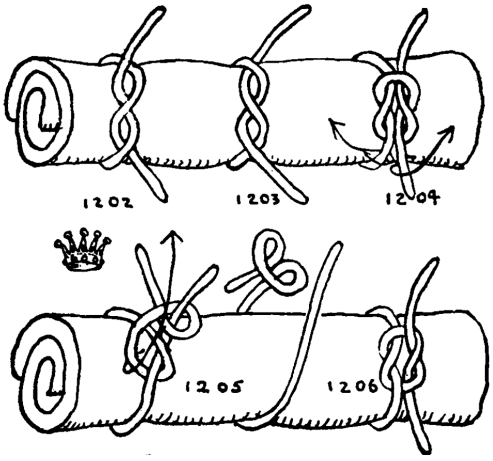


Figure 7: (1204) The reef knot, (1206) The granny knot. Reprinted from Ashley [1944, page 220].

Figure 8: (1843) The anchor bend. Reprinted from Ashley [1944, page 209].

to solve the problem in an automated manner, the idea of *physical environment augmentation devices* or PEADs has been proposed. The final design of such PEAD capable of linearly moving and rotating CSI acquisition device is depicted in Figure 13.

2.5.2 Knots

During the PEAD construction two types of rope binding tasks occurred:

1. binding of the two rope ends, usually under tension and
2. tying of rope end to the middle of another rope.

For the first task, the *blood* and *reef knots* were attempted.

- The *blood knot*, Figure 6 is considered to be one of the strongest knots Ashley [1944, page 259] [Wilson, 2003].
- The *reef knot*, Figure 7 (1204) is one of the simplest and universally used knots Ashley [1944, page 220] and can be tied while conserving target ropes original tension.

It is important not to tie a *granny knot* instead of the *reef knot*. The *granny knot* constitutes of two identical half-knots, Figure 7 (1206), as opposed to the *reef knot* from two opposite half-knots.

For the second task of tying a rope to the middle of another rope a comparatively simple Ashley [1944, page 209] *anchor bend*, Figure 8 has been tried.

3 Materials and methods

This chapter determines initial methods that have ignited the revolutions of the further project work, described in Results chapter. It begins with determining the software framework, libraries, and primary data storage structure in Section 3.1. Section 3.2 describes relevant properties of raw CSI data. Section 3.3 identifies the utilized CSI collection protocol. Lastly, Section 3.4, justifies the preference of Wi-Fi transparent materials for *physical environment augmentation devices* construction along with the incorporation of servo motor components.

3.1 Software

In order to avoid the burden of programming all the needed functionality to train and deploy a neural network model, a machine learning framework may be used. In particular, `Pytorch 1.3.1` is deployed within the scope of the present thesis. Other widely used libraries include `Numpy 1.18.1` for data processing and `Pandas 1.0.3` for metadata storage and sample segregation.

The present work trains neural networks using one of the implementations of back-propagation and stochastic gradient descent (SGD) algorithms [Goodfellow et al., 2016]. The loss and gradients for back-propagation are computed for each sample in the mini-batch. Then they are averaged over all mini-batch samples prior to application for performance reasons. Without shuffling, mini-batch contains subsequent samples from the train sample sequence. However, when shuffling is applied, mini-batch draws random samples without replacement. For example, without shuffling a mini-batch may contain 32 subsequent *"Kitchen"* labels, while with shuffling these would be *"Kitchen", "Lobby", "Living-room", "Empty",* etc. This decreases the correlation between samples in mini-batch and averages out non-general vectors.

The computer used for ML has three primary hardware methods to store raw measurements and preprocessed dataset – hard magnetic disk drive (HDD), solid-state drive (SSD), and random access memory (RAM). The large raw data could be stored only on HDD. Aside from being the slowest method for continuous data retrieval, HDD suffered from long random sample access delays presumably due to the need to re-position the mechanical reading head. RAM, on the other side, did not have enough capacity to store the whole preprocessed dataset. Taking into account HDD and RAM bottlenecks, filler CSI data between recorded actions was cut out, remaining samples preprocessed and stored on SSD in a form of a `numpy` memory map. `numpy.memmap` creates an `np.array` image on disk, which can be accessed and read by small parts, without loading the whole array into RAM. Such technique is especially useful together with the default `Pytorch` dataloader, which has an option for assembling mini-batch from randomly chosen samples without replacement.

3.2 CSI data

3.2.1 Multipath interference

As mentioned in Introduction, one of the methods to achieve environment agnostic classifier operation is to manually remove environment-specific information from recorded CSI data. An example of such environment-specific information is multipath interference. In fixed surroundings some subcarriers consistently experience constructive interference, while other, destructive. This results in a specific stable multipath configuration and rather constant correction coefficients offsets. Neural network trained with just these offsets gets used to them and fails to produce proper predictions when Wi-Fi device is moved or rotated, which leads to an offset distribution change. An example of the amplitude offset distribution change can be seen in Figure 9. It is, however, possible to normalize correction coefficients for each subcarrier with respect to mean and variance and supply NN with more environment agnostic information.

3.2.2 Amplitude and phase

As it can be seen from Figure 10, variations in both amplitude and phase behavior of complex subcarrier correction coefficients coincide in time. The fact that such differences are visible with naked eye serves as an evidence that both amplitude and phase contain at least some information about the conducted activity and hence can serve as inputs to neural network separately or in combination.

3.2.3 Delays between reports

Although the target CSI collection rate (the number of reports collected from each AP-STA link per second) is set to the maximum for the deployed CSI acquisition device AP of 100 Hz, the real collection rate varies. As it can be seen in Figure 11 (A), the delays between the reports from all 3 AP-STA links are distributed close to the Gaussian form, with the shortest delay below 3 ms, and the longest above 22 ms. In an ideal case, one might expect reports from 3 AP-STA links to be evenly mixed (coming in 1-2-3 pattern). However, even the number of reports from different links may vary by multiple times. For example, Link 3 yields 451 out of total 700 reports in the 5 s sample. Meanwhile, Link 1 yields only 93 CSI reports. This gives $93/451 \approx 1/5$ ratio in one randomly selected 5 s sample. Delays distribution for a whole recording of ≈156 seconds can be found in Appendix C.

During the data preprocessing stage, sliced samples for each link are extrapolated to a unified shape, matching the input shape of a neural network. Nevertheless, if a report with only few datapoints with long delays in between is extrapolated, it is likely to have a negative impact on the neural network training. Therefore, records lacking an empirically chosen minimum number of reports were excluded from the training dataset.

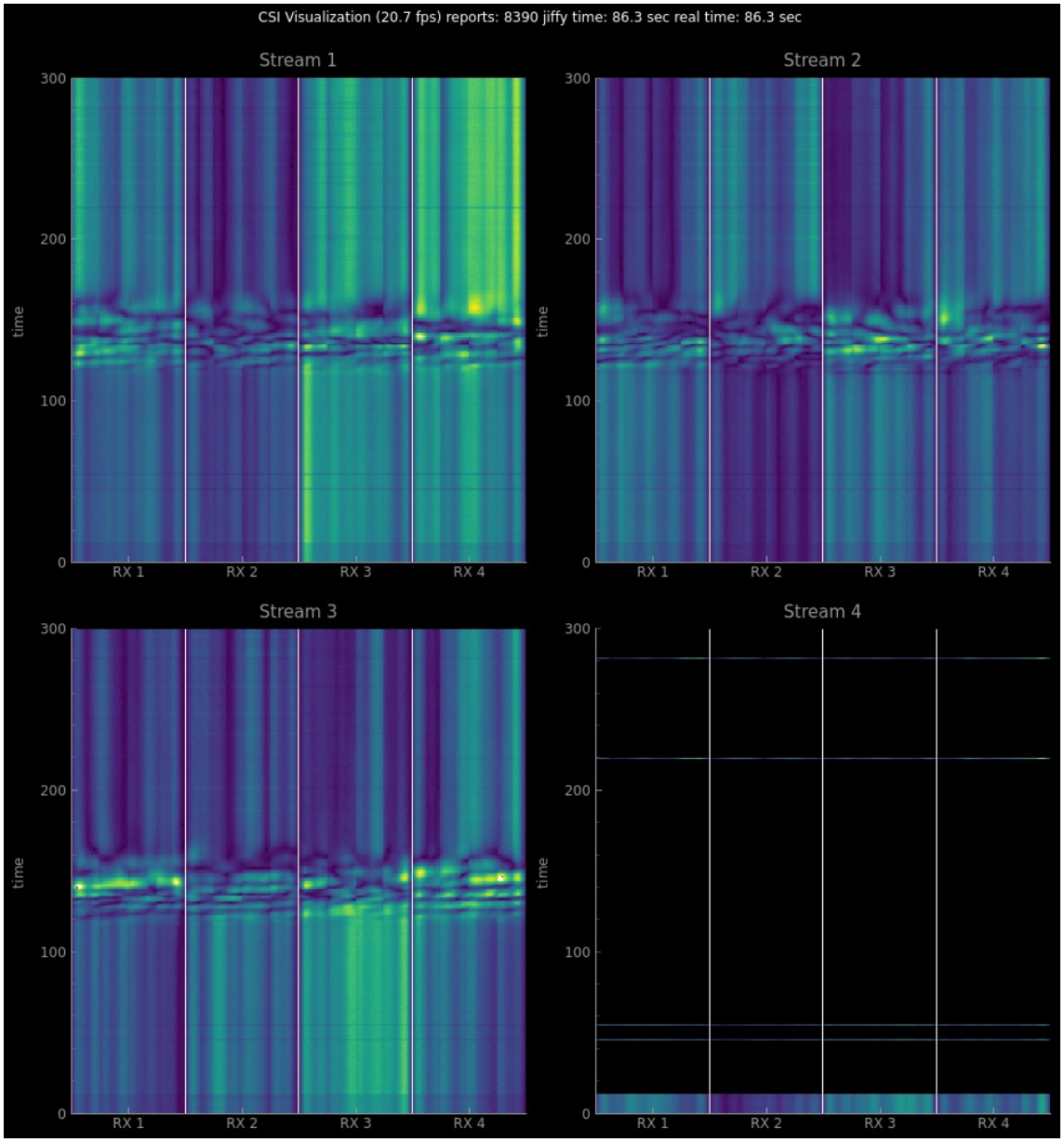


Figure 9: Channel state correction coefficients visualization for 3 seconds; the vertical axis is for time, horizontal is for subcarriers. AP and STA are stationary until the STA is rotated by 90 degrees between seconds 1 and 2. This causes a change in the multipath configuration and a subsequent change in subcarriers correction offsets. Similar to the observed switch happens not only after rotation, but also after STA or AP movement or even after noticeable transmission channel change, for instance, after placing a body part right next to the device's antennas.

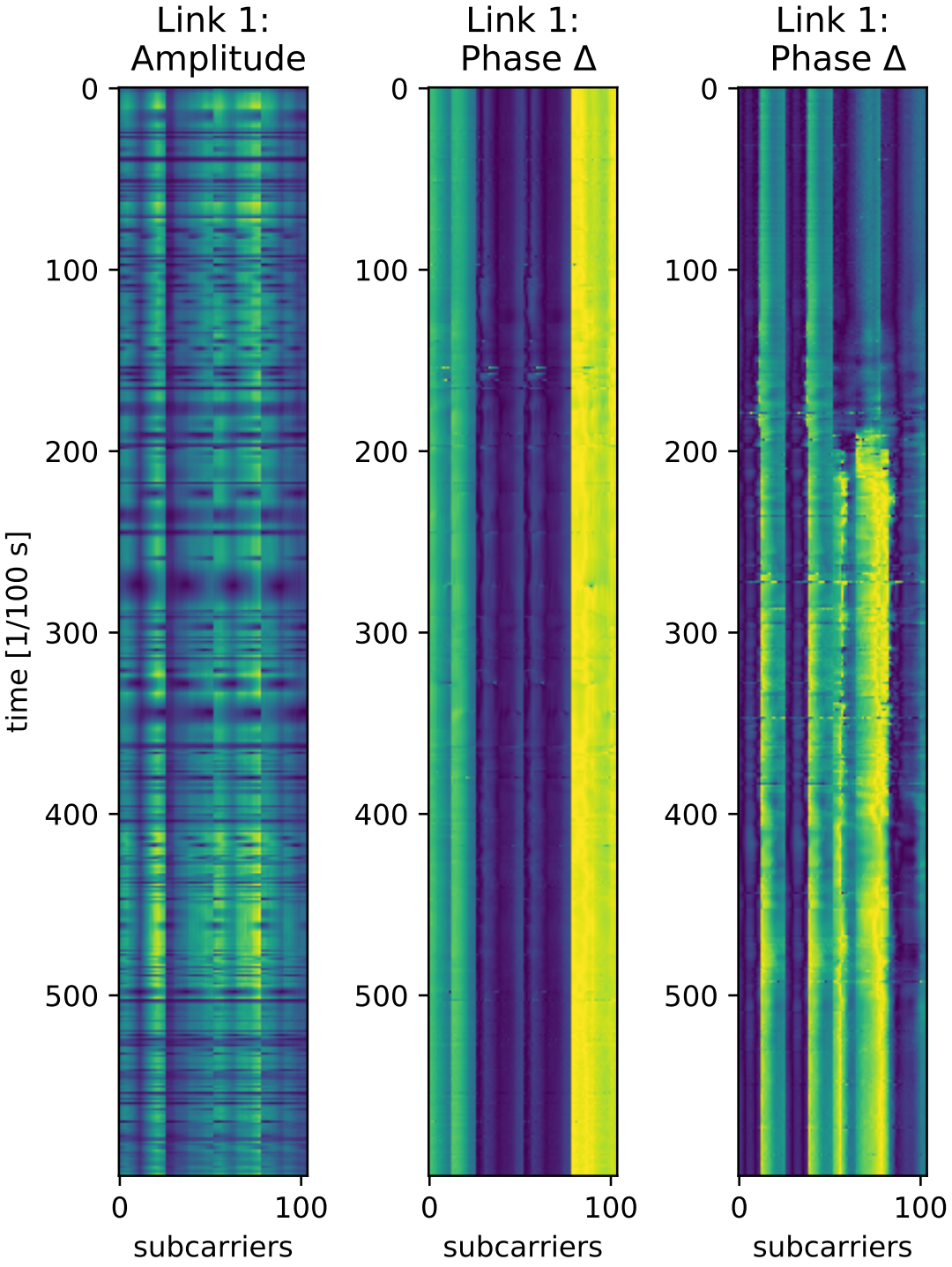


Figure 10: Six seconds CSI recording of a "Lie down" activity. Left subplot: CSI amplitude. Amplitude is visualized for $(Tx[1,2,3,4], Rx[1])$ paths. Each (Tx, Rx) path yields 26 subcarriers, which concatenate to 104 subcarriers in the horizontal axis. Right and center subplots: phase difference between $(Tx[1,2,3,4], Rx[1])$ paths and another randomly selected $(Tx[1,2,3,4], Rx[2\ or\ 3\ or\ 4])$ paths. It can be noticed that visualized relative phases are different. Nevertheless, some signatures of "Lie down" activity are visible in both phases plots.

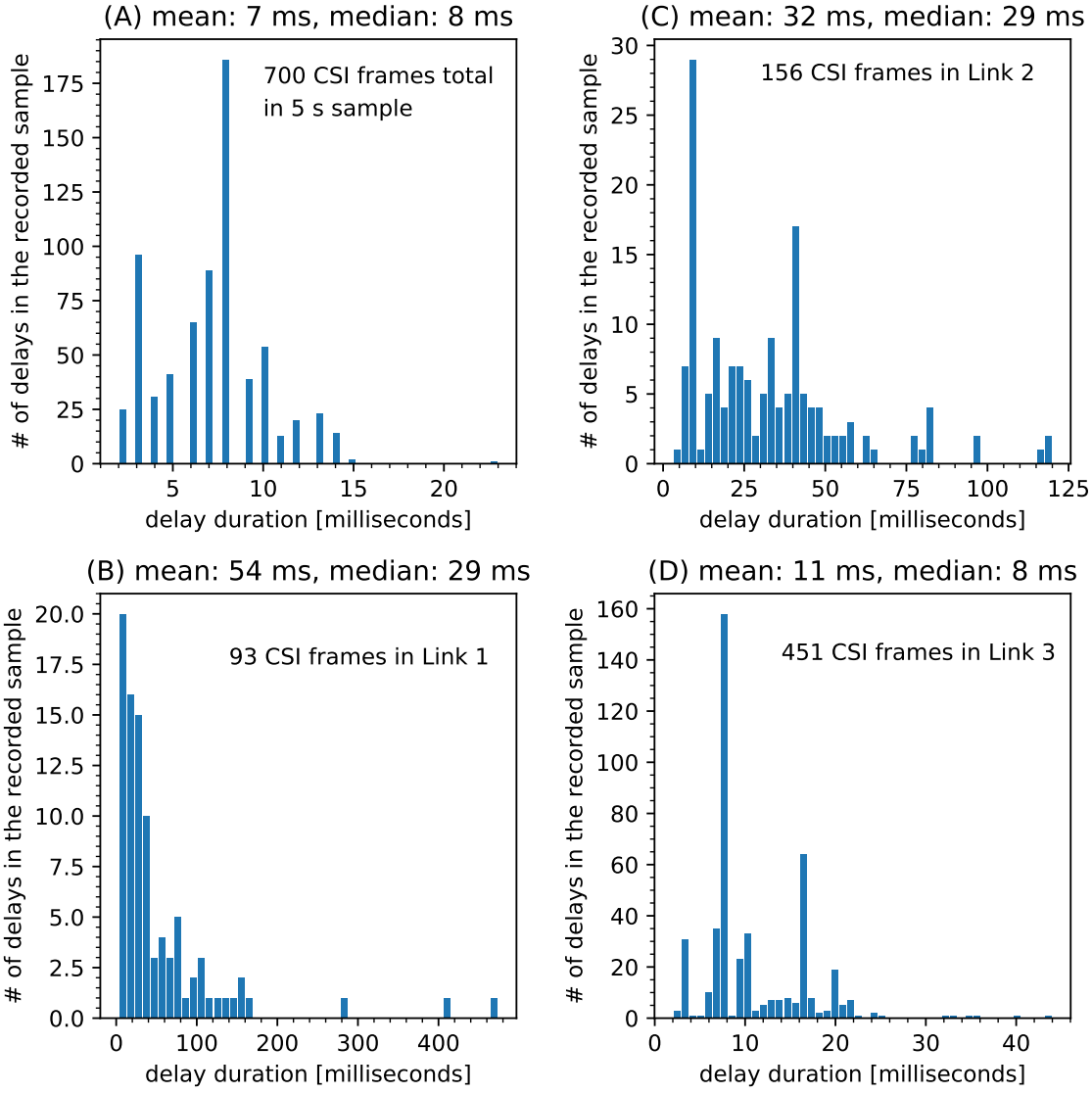


Figure 11: Delays distribution between subsequent CSI reports. (A): Delays between reports from all 3 AP-STA links, zipped in a single stream. (C-D): Delays between individual AP-STA link reports. Please note that scales on the histograms are different and emphasize acute divergence in the reporting rate from different links. The number of reports acquired from each link varies as well – link (B) contains 13% of sample reports, link (C) – 22%, while link (D) – 65%.

3.3 Data collection routine

CSI data collection is organized as follows:

1. Researcher writes audio-scripts and generates audio instructions for test subject(s) to follow. Examples of audio-scripts can be seen in Appendix B.
   (a) Audio-scripts for human activity classification describe the sequence of activities the test subject has to perform.
   (b) Audio-scripts for human localization specify the location the test subject must stay within, as well as the activity to continuously perform in the given location.
2. Researcher prepares a data collection plan, which includes at least the following information:
   (a) location(s) of experiments,
   (b) list of recording hardware,
   (c) audio instructions to be used,
   (d) number of test subjects,
   (e) data collection curriculum for each test subject, and
   (f) PEAD positions for each curriculum cell.
3. Location is prepared for data collection:
   (a) PEADs are assembled and placed at specified locations.
   (b) Functional furniture is brought in. For example, chairs for sitting down / standing up activities, mats and beds for lying down / getting up activities.
   (c) Walking passages are freed from obstacles.
4. Hardware is tested.
5. Test subject (which could be the same person as the researcher) follows audio instructions and performs requested activities at the specified location.
6. At the end of each CSI data recording session at least the following information is automatically uploaded to the cloud storage:
   (a) recorded CSI data,
   (b) activities / localization labels (see Appendix B for examples) and
   (c) measurement metadata, which includes:
      i. measurement location,
      ii. test subjects IDs,
      iii. list of used sensors, and
      iv. measurement ID and save location.

7. In addition to the minimal mandatory information, arbitrary sensory data can be binded to the measurement. For example, measurements conducted and analyzed in the present work are accompanied by video records.

3.4 Physical environment augmentation devices

3.4.1 Preference for Wi-Fi-transparent materials

One of the design goals for the PEAD was to minimize the Wi-Fi signal obstruction. Therefore, the usage of highly conductive metal parts [Cheng, 2018] had to be minimized. Consequently, preferred commercially available construction materials were plastic, wood, and cloth.

3.4.2 Servo motors

Compared to standalone motors, servo motors embed many necessary elements in a single package:

- motor itself,
- gears reducer,
- angular position feedback sensor,
- control commands decoder, and
- motor driver circuit.

Since the development speed was one of the priorities for PEAD design, standalone DC servo motors were chosen to power the mechanical elements of the construction.

An additional benefit of many low power servo motors is that they operate on 5V DC, which is the same for both RPi, deployed Wi-Fi AP and STA. This opens the possibility to power all the mandatory PEAD electronics from a single voltage source. At the same time, many commercially available power banks supply 5V DC as well, and hence provide a straightforward way to turn PEAD into a mobile, socket-independent device.

The default RPi GPIO library outputs an unstable pulse-width modulation (PWM) signal, which results in a jitter and undesired position shifts in servo motors. In order to avoid these default library drawbacks, the specialized PWM library utilizing direct memory access (DMA) `pigpio` has been used.

4 Results

This chapter begins in Section 4.1 with the development of *physical environment augmentation devices* aimed to facilitate with the Wi-Fi CSI data collection. It then proceeds with establishing new physical data collection spaces in Section 4.2. After activities data collection, the chapter describes the results in human body localization neural network size reduction (Section 4.3). Soon after, it returns to the topic of environment agnostic activity classification in Section 4.4 and summarizes this area results.

4.1 Physical environment augmentation devices

The purpose of a *physical environment augmentation device* is to automatically displace and rotate a Wi-Fi CSI acquisition equipment in space to change the multipath configuration of subcarriers.

4.1.1 End result overview and components description

The design of PEAD has undergone three iterations. The first prototype can be seen in Figure 12. The third and final iteration – in Figure 13.

Below is the high-level explanation of PEAD's components' functions. Numeration follows the labels in Figure 13:

(1) **PVC pipes**. As explained in Section 3.4.1, in order to minimize RF interference, the size and number of metal parts had to be minimized. At the same time, unlike wooden planks, PVC pipes could be easily joint and disjoint. This has been considered helpful for postage packing and subsequent on-site assembly.

(2) **Guide ropes**. Without extra guidance, payload and counterweight would have been swaying after any horizontal movement of the PEAD. With two guiding ropes on each side, payload and counterweight slide along the guides with negligible oscillation.

(3) **Wire connectors**. All long wires can be easily disconnected, which is helpful for transportation.

(11) **The height-adjustment wheel**. Rotation of this wheel causes the payload and counterweight to change height in opposite directions.

(12) **Continuous rotation servo motor**. Powers the height adjustment wheel.

(13) **Rubber bands with hooks**. Used as anchor points for guiding ropes. Rubber creates constant force to keep guiding rope strained. Bands are wrapped around the horizontal wooden plank, making it possible to slide them sideways thus changing width between guiding ropes to conform with payload and counterweight thickness.

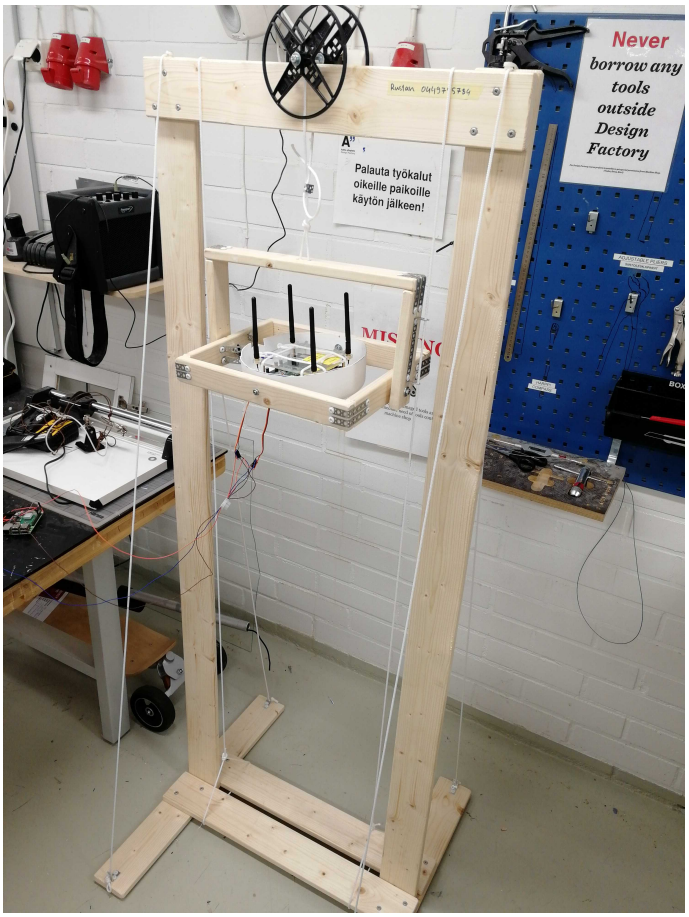


Figure 12: The first iteration of the *physical environment augmentation device* (PEAD) design.

(14) **Secondary metal wheel**. Payload and counterweight are rather wide and could be lifted up and down by either one large-diameter wheel, or two smaller wheels. Smaller wheels consume less vertical space and are cheaper than a large-diameter one. Another role of a secondary metal wheel is to short-circuit the electric contour and provide feedback about the vertical position of the payload.

(15) **Upper wooden structural beam**. Prevents PEAD collapse from forces acting in a sideway direction. Beam is attached at 45° angle and kept as long as possible without interfering with height-adjustment wheel.

(16) **Wires**. Continuous rotation servo wires and electric feedback contour wires in a screw connector.

(17) **Twisted wires**. These wires suspend payload and counterweight. At the same time, they serve as a part of electric vertical position feedback contour.

(18) **Power extender**. Hosts power supplies for RPi, servo motors, and Wi-Fi acquisition device.

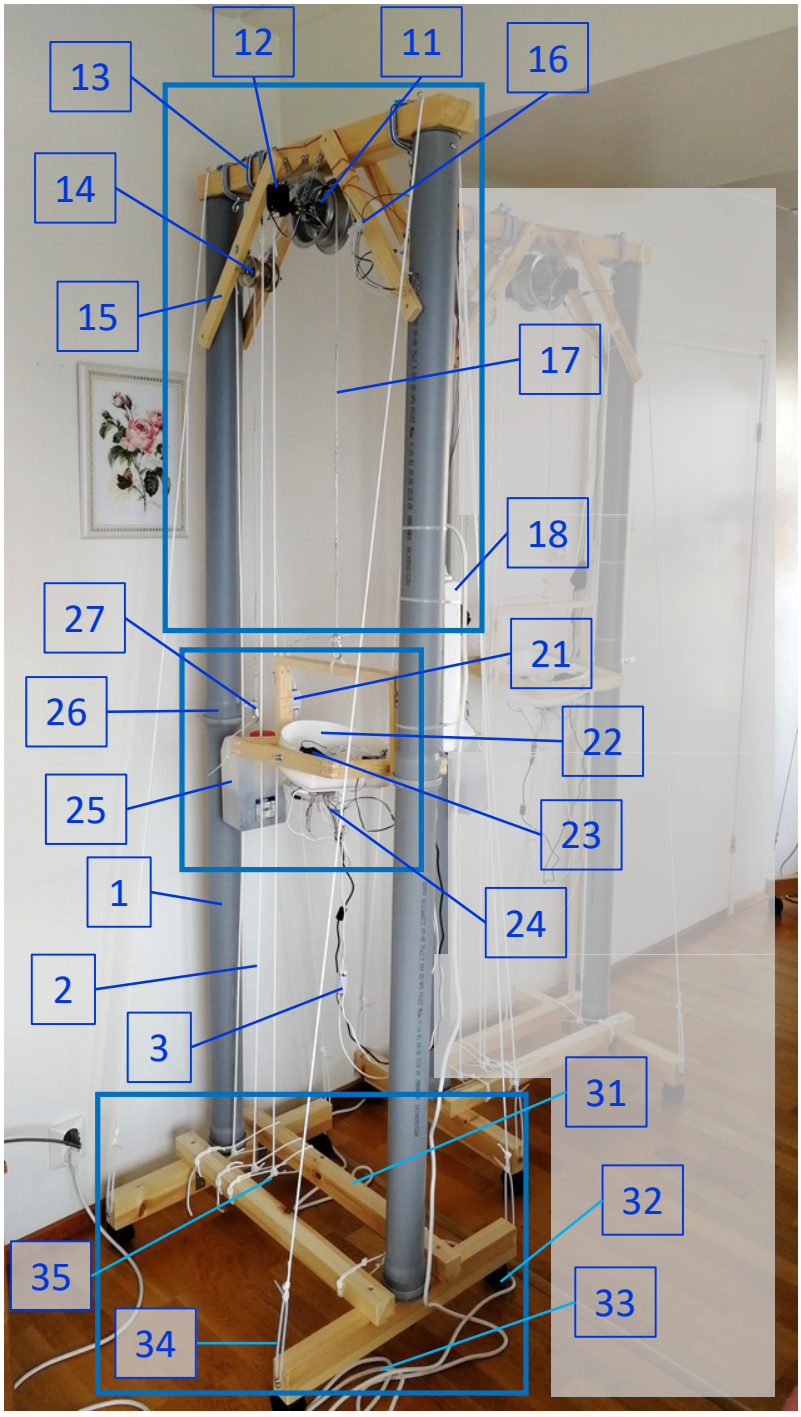


Figure 13: The final design of *physical environment augmentation device* (PEAD). Label explanations are in the main section text.

(21) **8P8C female-female connector**. Serves as a decoupling point for electric feedback contour during PEAD transportation.

(22) **Plastic hull**. Contains a Wi-Fi device, a RPi and an inner servo motor.

(23) **Inner servo motor**. Two of such motors placed at 90 degrees to each other, can rotate the hull to an arbitrary angle.

(24) **Wires from RPi GPIO**. These wires interface servo motors as well as electric feedback contour to the Raspberry Pi.

(25) **Counterweight**. A plastic canister filled with sand which weight is approximately equal to the weight of the payload (AP or STA, RPi, servo motors, the hull, and wooden planks around it).

(26) **Pipes joint**. Connects upper and lower PVC pipes. May be disjoint during PEAD transportation.

(27) **Clasp**. Counterweight and payload can be attached to twisted wires with clasps.

(31) **Lower wooden structural beam**. Beams are placed wide enough to prevent base collapse from potential sideways forces and kicks. On the other hand, they are placed narrow enough to serve as anchor points for the guiding ropes without pulling the connection point above beams.

(32) **Plastic wheels**. Two front wheels have stoppers, so that PEAD frame can be fixed in space.

(33) **Extra power cable**. 5 m power extender allows some PEAD movement without re-plugging.

(34) **Tightening clip**. Ropes that prevent PEAD fall and adjust PVC pipes frontal inclination can be tightened individually using plastic clips.

(35) **Base rope loops**. Loops act as bottom anchor points for guiding ropes.

In total, seven PEADs have been constructed to facilitate the CSI data collection. An easy on-site assembly was one of the design goals for the *physical environment augmentation device*. The final design takes ~15 minutes for a trained person to unpack and assemble from the transportation state. To facilitate such training, 31.5 min of video instructions covering all aspects of assembly have been filmed.

4.1.2 Vertical position electric feedback contour

Different environments create different Wi-Fi multipath configurations, which result in different subcarrier distributions (Figure 1). In order to emulate different environments, Wi-Fi AP or all STAs may be rotated (Figure 9) or displaced.

The rotation of a device is performed with servo motors. As described in Section 3.4.2, servo motors contain the angular position feedback sensor, which allows them to reach and maintain the target angle. Once calibrated, they receive PWM signal encoding the target angle from RPi and use in-build feedback loop to maintain it.

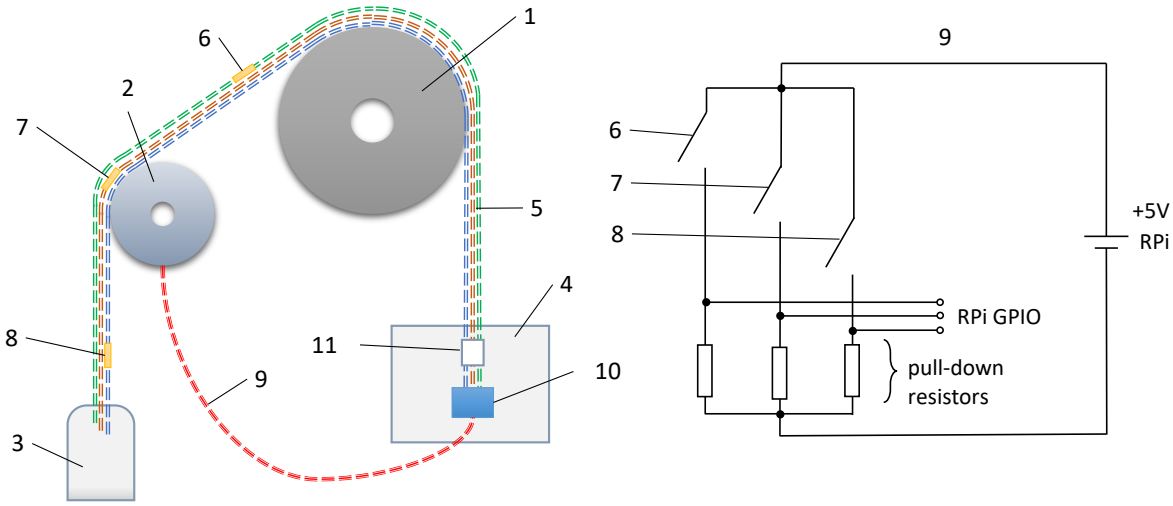


Figure 14: Vertical position feedback contour. *On the left*: Physical components. The servo-powered height-adjustment wheel (1) moves the counterweight (3) and payload (4) in the vertical dimension. The counterweight and the payload are suspended on twisted wires (5) with peeled insulation segments (6,7,8). When the exposed core of a twisted wire touches the secondary metal wheel (2), an electric contour looped by wire (9) is closed. This is sensed by a pin of Raspberry Pi (10) and a momentary vertical position of the payload is determined. In order to make the apparatus more transportable and modular, twisted wires are attached with an Ethernet connector (11). *On the right*: An equivalent electric principal schematic.

Unlike the Wi-Fi device rotation, its vertical displacement powered by continuous servo motors required a separate height feedback loop. Such *electric vertical position feedback contour* is described in Figure 14.

The effective vertical range, or distance between top and bottom position of the payload is 140 cm. As there are only three positions in total, an average distance between them is 70 cm. Such vertical displacement is deemed sufficient to change multipath configuration and hence amplitude distribution of Wi-Fi subcarriers.

4.1.3 PEADs technical solutions

During the PEAD development stage each component has been thought-through and iterated several times in order to achieve easier construction, transportation, on-site assembly, lower price or additional functionality of the device. Some of design findings are described in the present section.

Payload hull and wiring. The payload (parts 21-24 in Figure 13 or part 4 in Figure 14) is designed to rotate around two perpendicular axes in a gyroscope manner. One may consider rotating sphere as an optimal gyroscope inner body shape for minimizing payload's (and hence whole PEAD's) dimensions. Since relatively to the inner hull, one servo motor has to be outside, its protruding has been compensated by

deploying bottom 10 cm of a wall-mounted bucket (Figures 16 and 18). Such cut-out white plastic hull hosts a Wi-Fi AP or STA inside along with an RPi screwed to the bottom. Such arrangement achieves less than a centimeter gap towards the thin wooden payload frame during rotation. The wiring to rotating parts is performed to ease the movement while working reliably over time, as shown in Figures 15 and 16.

Ethernet connectors. The *electric vertical position feedback contour* as well as servo motors wiring is conducted with AGW24 single core wire. Such wire has been found to fit in a standard 8P8C Ethernet connectors – as depicted in Figure 17. Adding such connectors at pivot points proved to be an easy and rather reliable way to simplify PEAD's disassembly and transportation (Figure 18).

Height-regulating servo motor arrangement. The rotations of the height-regulating wheel and continuous servo motor (Figure 21) are not completely coaxial due to high tolerances of manually crafted components and overall PEAD assembly. If the wheel and the motor are firmly attached to the wooden frame and rigidly fixed against each other, the inevitable difference in axes alignment would cause a significant force to be exhorted on the rotor shaft. Such force could cause fast wear-off of servo gears and might break the motor over time.

In order to decouple the wooden frame, the servo motor and the wheel, metal stripes in specific configuration have been applied. First of all, the frame-to-motor stripe (quarter-circle stripe in Figure 21) allows the motor body to move slightly sideways rather easily while counteracting its self-rotation. Secondly, as more closely shown in Figure 20, the interface between the servo motor and the wheel has a bending joint to further decouple misaligned rotation axes.

Height-regulating wheel. The wheel is a grey-painted rim, attached to the wooden frame with an arrangement of metal stripes (Figure 22). Two metal stripes that the wheel is hanged on, form a rigid triangle with the wooden frame. Such arrangement prevents the wheel from shifting sideways when horizontal forces are applied. The central hole of the rim intended for a shaft has a much larger diameter compared to holes in metal stripes. Therefore, a shaft from a long screw of the proper diameter is improvised (Figure 23). The shaft is attached to the wheel with the curved metal stripe from each side and rotates together with a wheel. Hence, if a rotor of a servo motor is attached to the shaft, it will rotate the wheel as well.

The paint covering height-regulating wheel, although not entirely necessary, provides an insulation from exposed contacts of the electric feedback contour.

Guiding ropes pass through vertically travelling counterweight and payload, (Figures 27 and 28) and prevent them from colliding and swinging. In order to create necessary tension, guiding ropes are attached to rubber bands via hooks at the top of PEAD, as seen in Figure 24. In previous iterations of PEAD design, rubber bands were not used but normal ropes with more intricate *blood knots* (Figure 6). After ropes were replaced with thick rubber bands, knots were changed accordingly to more suitable *reef knots* (Figure 25). As for the bottom, guiding ropes are tied to lowest parts of rope loops via *anchor bends*, as can be seen in Figure 26.

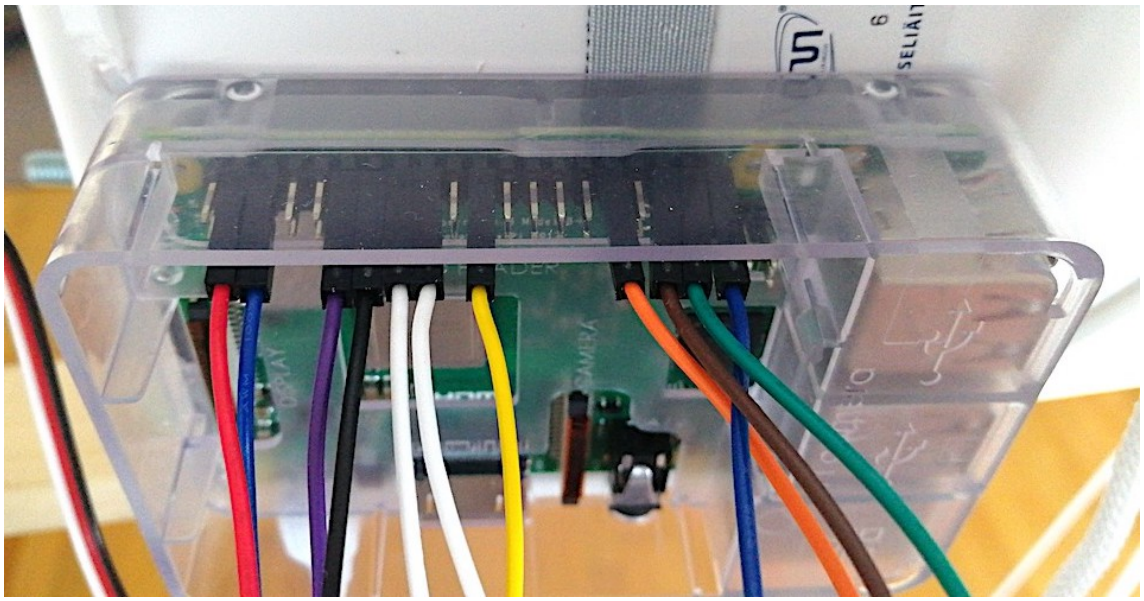


Figure 15: All GPIO outputs from RPi have been configured for upper pins row only. This arrangement allows pouring hot glue in the space between pins and the upper wall of the transparent RPi case. Hot glue prevents wires to fall from GPIO pins during long-term operation.

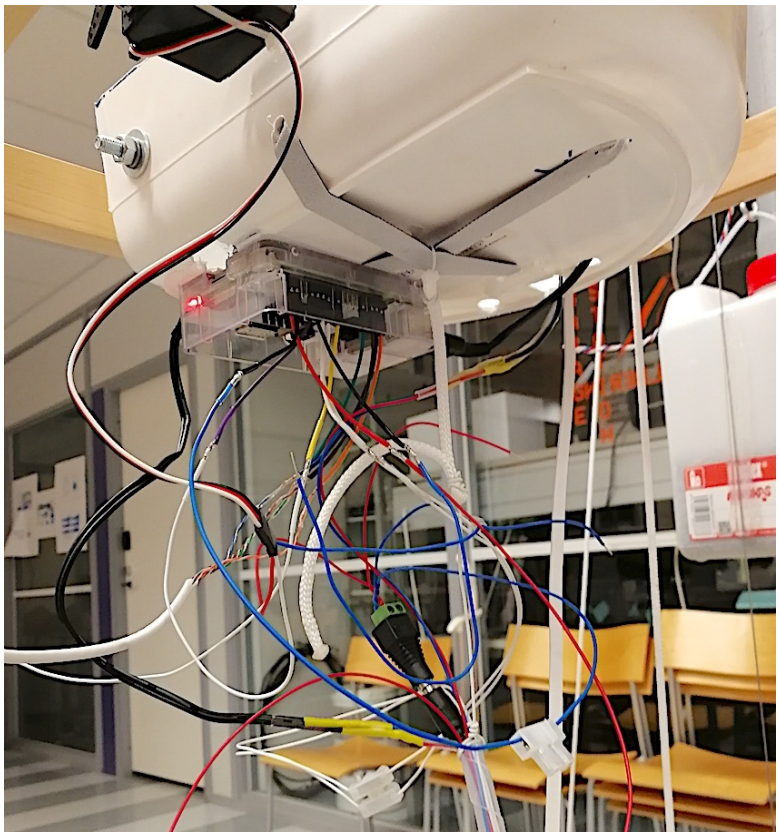


Figure 16: Wires connected to RPi are intentionally sparsed out in order to ease the rotation of white hull for servo motors. Note that soldering joints between RPi jumper wires and single-core breadboard wires are insulated with transparent thermoshrinkable tubes. All wires coming to the payload hull pass through connectors which allows payload convenient separation from the whole device if needed.

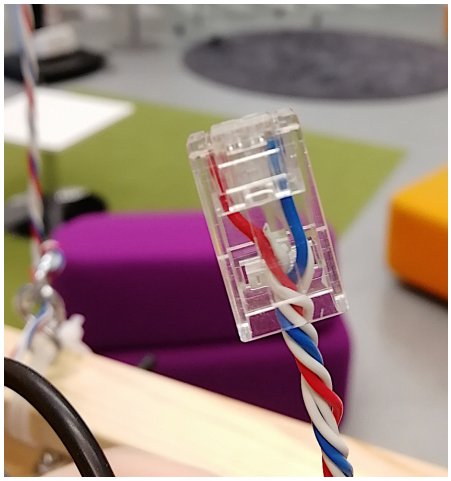


Figure 17: Single-core breadboard wires of *electric vertical position feedback contour* are interfaced with the 8P8C connector.

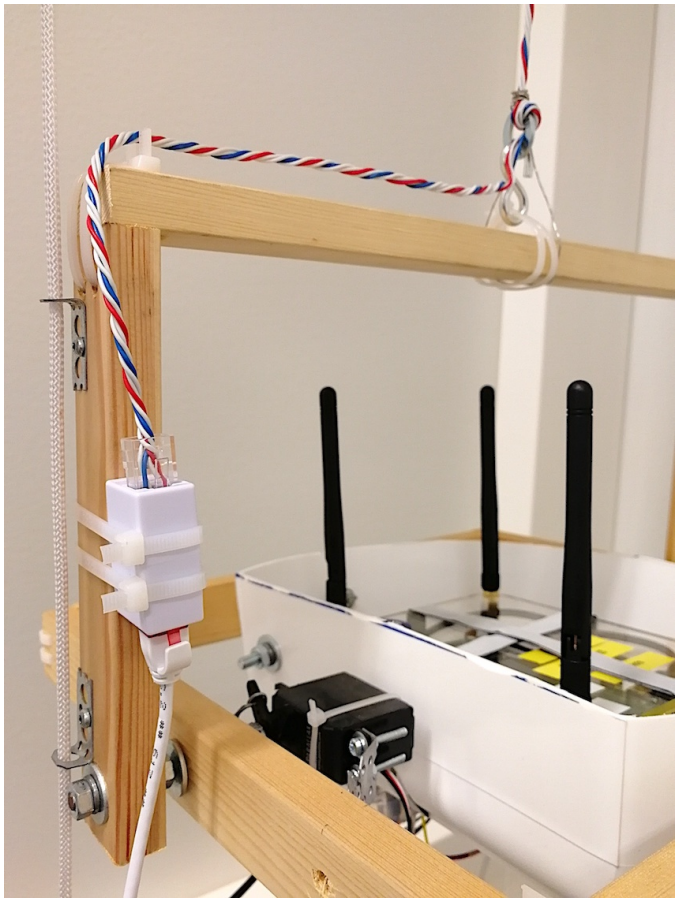


Figure 18: Side view of the payload. In order to allow payload to be disconnected from the device during transportation, wires from RPi screwed to the bottom of the white plastic hull are separated from the rest of the *electric vertical position feedback contour* by an Ethernet female-female connector.

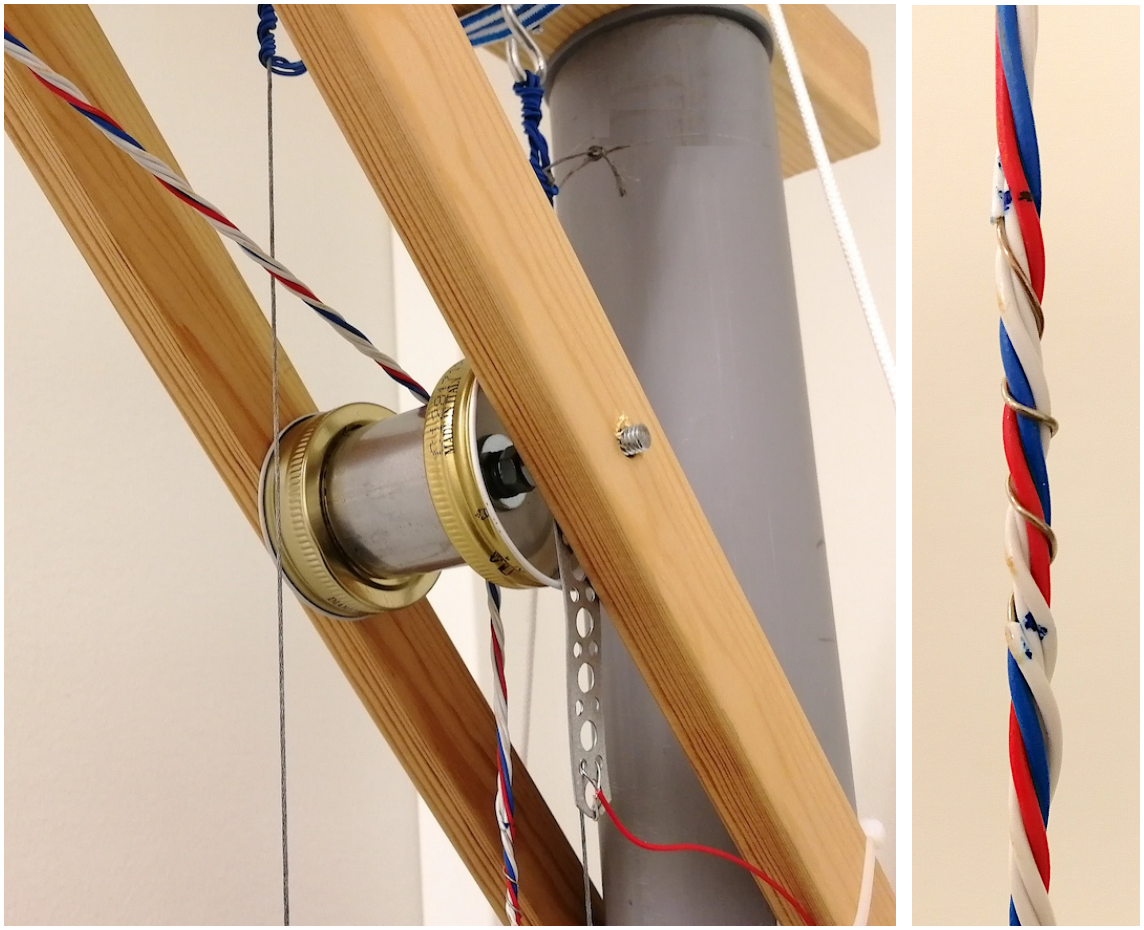


Figure 19: Secondary metal wheel that shorts the *electric vertical position feedback contour*. When regions of twisted wires peeled from insulation (on the right) touches the surface of an improvised wheel, the contour circuit is shorted. In the final implementation, three peeled insulation regions are present, providing electric feedback for top, bottom, and middle payload vertical positions. Originally, feedback contour wires were based on Ethernet cables with soldered naked wire offshoots. However, soldering joints turned out to be weak points due to an abrupt mechanical impedance step between hard solder blob and an easily bending wire. Therefore, a solderless approach has been developed (on the right).

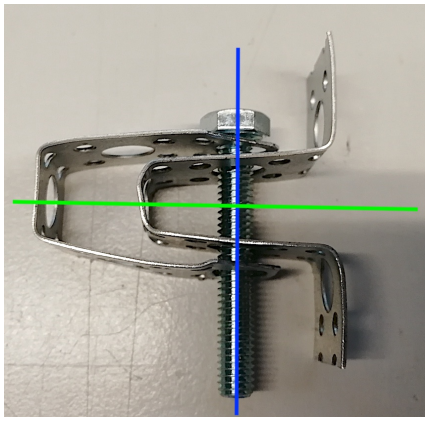


Figure 20: The interface between the continuous servo motor and the height-regulating wheel. The rotation is around the horizontal axis, outlined green. Two metal stripes can bend around the screw joint – vertical axis, outlined blue.

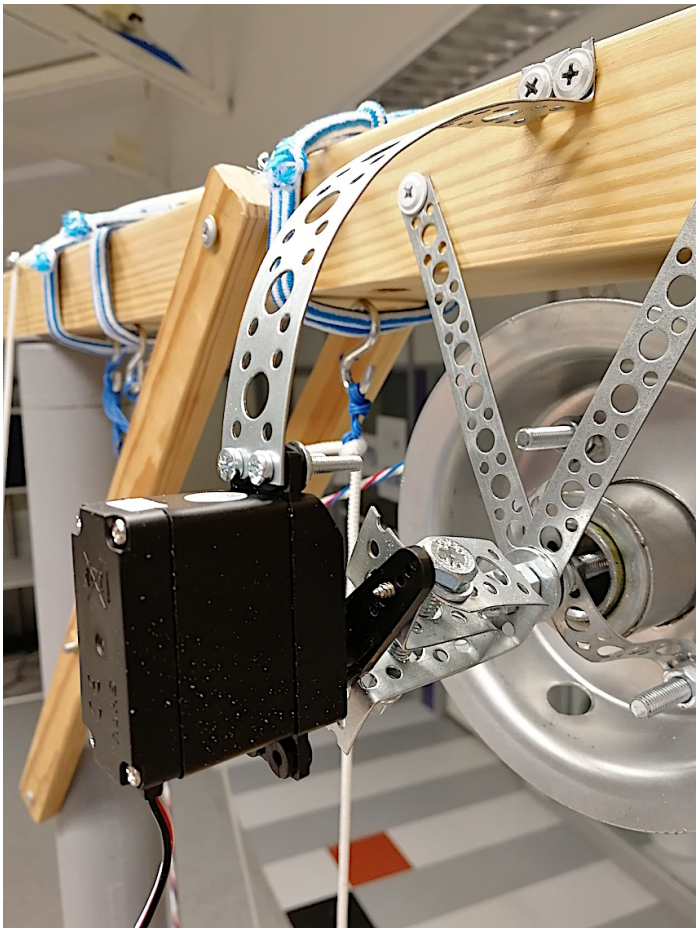


Figure 21: Arrangement of continuous servo motor attached to the height-regulating wheel. Rotation of the wheel causes payload and counterweight to change altitude in opposite directions. The motor is attached to the wooden frame with the quarter-circled metal stripe. Rotor to wheel attachment is done with two nested metal stripes.

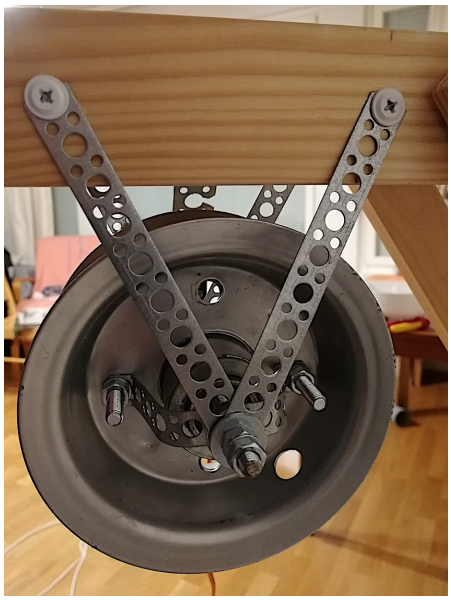


Figure 22: Front view at the height-adjustment wheel. The wheel is hanged from the wooden frame by two metal stripes at each side.

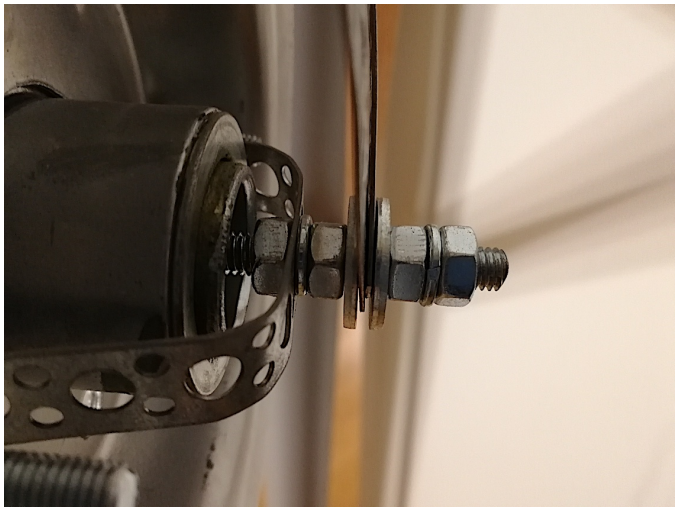


Figure 23: Side view at the height-adjustment wheel. The first two nuts from the left fix the threaded shaft to the rest of the wheel. The distance between two middle nuts is enough to allow the loose rotation of the shaft and the wheel with respect to the metal stripes it is hanged on. The last two nuts serve as a stopper and prevent the metal stripes to slip off a threaded shaft.

Figure 24: A hook to fix a guide rope on. Located at the top wooden frame beam.

Figure 25: *Reef knots* for rubber bands. View from above the disassembled upper PEAD section.

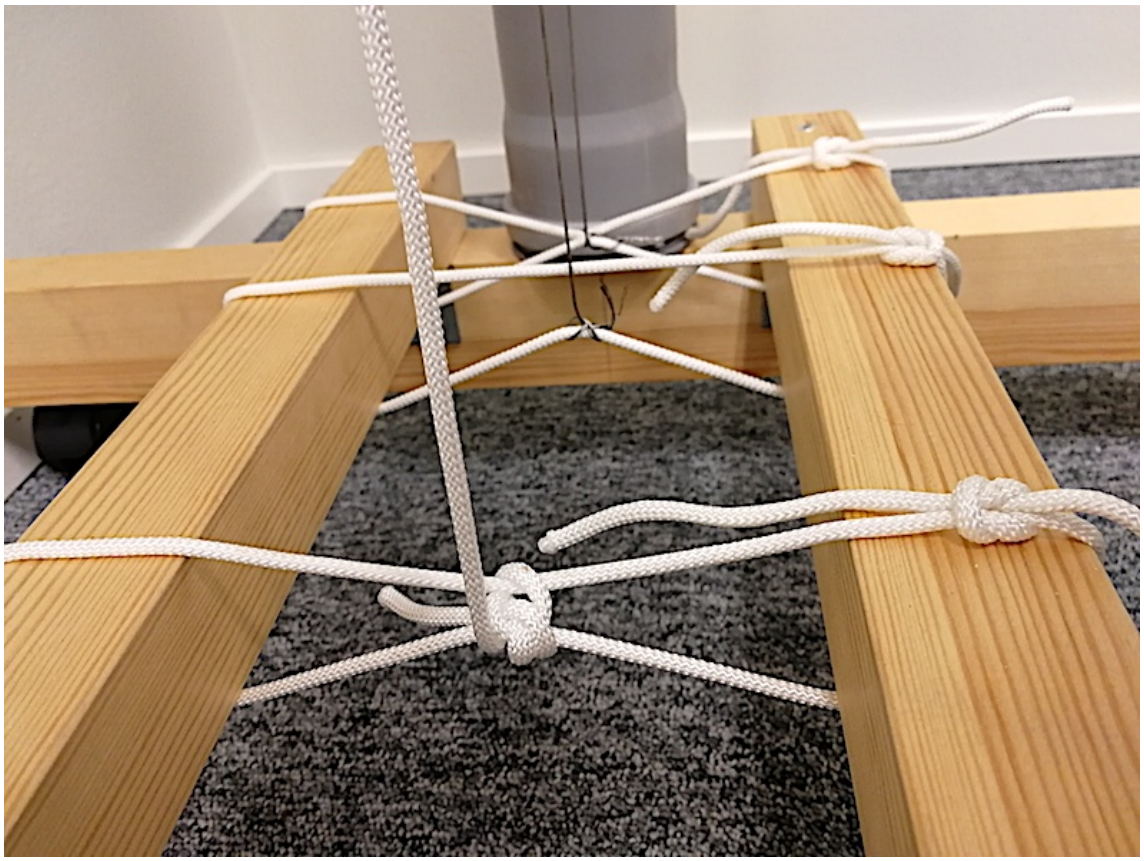


Figure 26: At the base of PEAD guiding ropes are attached to rope loops. The guiding rope may be tied to an upper part of a loop, or to a lower part. Tying a guiding rope to the lower part of a loop (knot on the middle loop) and subsequent wrapping it around (closest and furthest loops) causes the attachment point to descend below wooden planks surface. This maximizes an effective range of payload and counterweight horizontal swing.

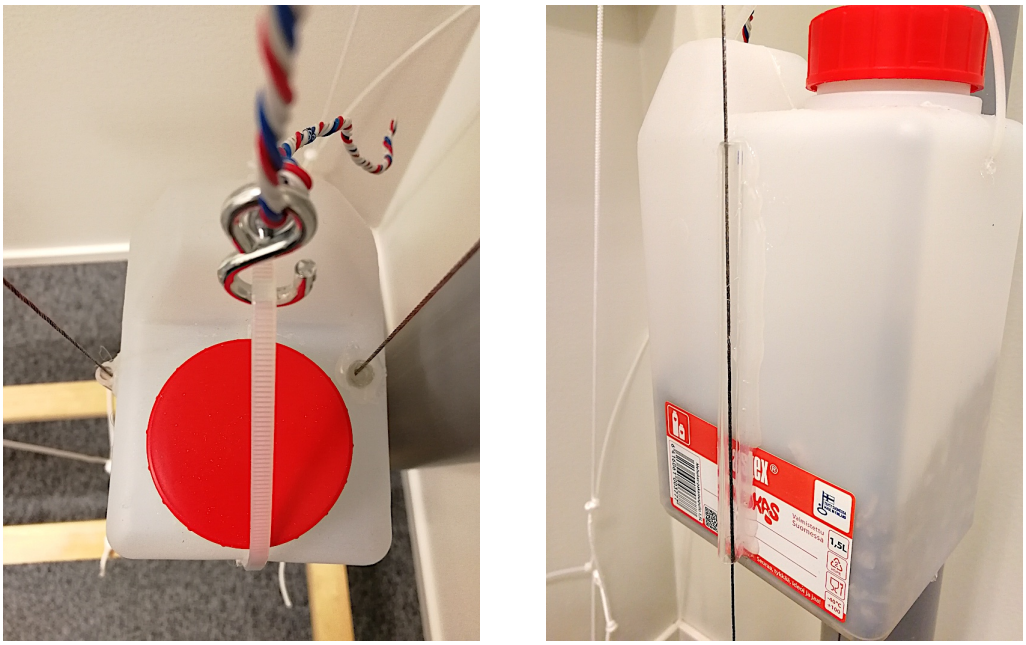


Figure 27: PEAD is relatively constrained in width dimension. Therefore, the right guiding rope passes directly through the counterweight canister. Additionally, exposing a rounded canister corner to the structural grey PVC pipe prevents counterweight seizing in case it hits the joint of structural pipes (Figure 13 marker #26). Both outer and inner guiding ropes are confined within plastic pipes, which prevents the canister fine rock filler to interfere with a guiding rope or pour out of the tank.

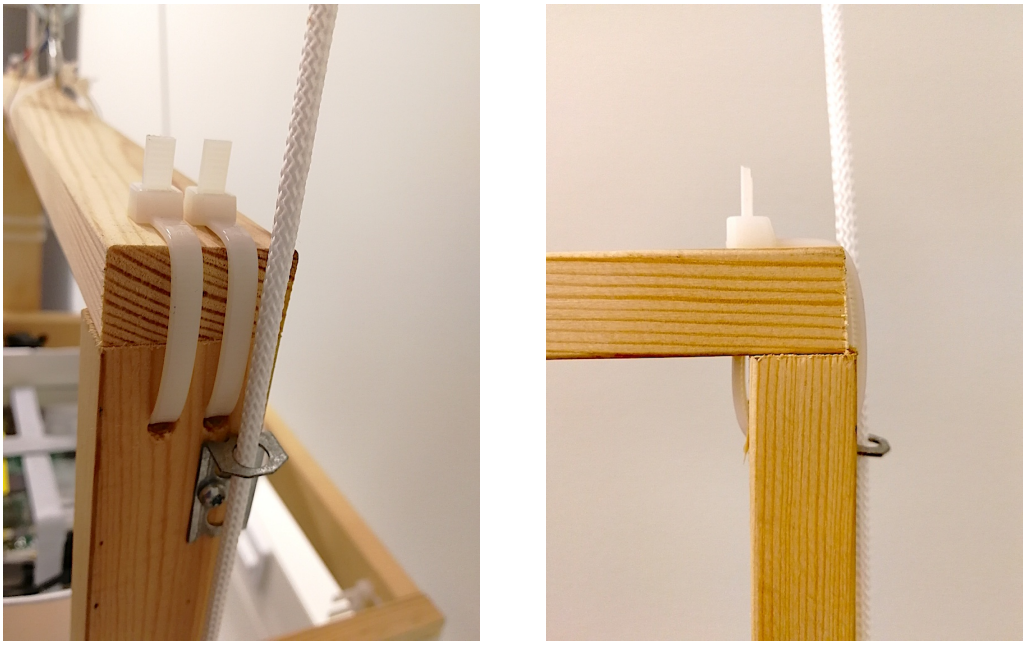


Figure 28: Guiding ropes of the payload are displaced from the wooden plank's center to prevent interference with a rotation joint. Wooden planks of a payload frame are held together with plastic tightening clips. This solution turned out to be virtually indestructible with bare human hands. Even if joint was dislocated after applying considerable manual force, it could be reversed and set back to the original position.

Figure 29: Assembled *physical environment augmentation devices* tested before shipment to the recipient countries. The second from the left device is compressed to the transportation mode.

4.1.4 PEADs summary

Each of the two payload servos are programmed to rotate it at +90, 0 and -90 degrees. The number of height positions is also 3 – top, middle and bottom. Together they yield 27 different Wi-Fi device positions with different subcarriers multipath configurations (Figure 9). When all positions are elapsed, PEAD can be moved horizontally by few meters and provide another 27 unique multipath configurations. Functionally PEAD operation allows a semi-automatic CSI collection from artificially diverse or "augmented" environments.

All together, transfers between two development locations – Aalto Design Factory and Prototyping Garage, as well as numerous visits to hardware and construction stores add up to 1441 driven through kilometers. The total time taken by PEAD development, manufacturing and transportation has been recorded to be ~596 hours. Figure 29 depicts devices prior to posting towards foreign CSI data collection locations.

4.2 CSI data collection in Espoo

Some data used in the present work has been collected in Sunnyvale, United States and Shanghai, China. In addition, two distinct collection spaces were created in Espoo, Finland.

The first space code-named "Merlin" is located in the Human Wellness Lab human testing facility. In order to keep people well confined within the 5×6 meters space (Figure 30) the chamber is surrounded with fine metal-layered walls. Aside from a remarkable sound insulation, said walls reflect and attenuate Wi-Fi signals by ≈20dB (Figure 32), thus creating a unique CSI collection environment. The place is schematically reflected in Figure 31. All together seven people have agreed to undergo the CSI collection of sitting and lying activities within the premises of Human Wellness Lab "Merlin".

The second data collection space has been chosen as an opposite to the Human Wellness Lab. This vast open space code-named "OpenOffice" is depicted in Figure 33 and cartographed in Figure 34.

Within the established "Merlin" and "OpenOffice" spaces the 3-links CSI data has been collected according to the protocol described in Section 3.3. Activities "Sit down", "Stand up", "Lie down" and "Get up" have been performed according to audio instructions similar to the one in Appendix B, and video-recorded to ensure correctness.

Activities datasets available for NN training after Espoo collections are summarised in Table 1. Datasets from Sunnyvale, Shanghai and Espoo have been recorded from different people. At the same time, Espoo datasets feature common participant which enables cross-environment and cross-people validation.

4.3 Human body localization

In the activity classification scenario a default humanoid body is expected from user. This similarity of the sensed object is a strong prior which allows a hope

Prior furnishing, lying on a yoga mat

After furnishing, lying on a bed

Panoramic view of the data collection chamber

Figure 30: Human Wellness Lab "Merlin" with metal-layered walls. In order to improve the CSI data diversity, sitting and lying spots have been re-positioned randomly during recording pauses. In addition, after collecting about half of the dataset, the room has been further furnished (upper pair of images). Note that after furnishing, "LIE_DOWN" label has been recorded while performing lying down on a bed rather than yoga mat. Bed continued to be randomly displaced after the furnishing. The lower image depicts the collection chamber configuration after the era of CSI activities collection has ended. Some equipment has been removed (two STAs in corners) while other – displaced (the STA in the left lower corner of panoramic image has resided above door during data collection, as on the upper two images).

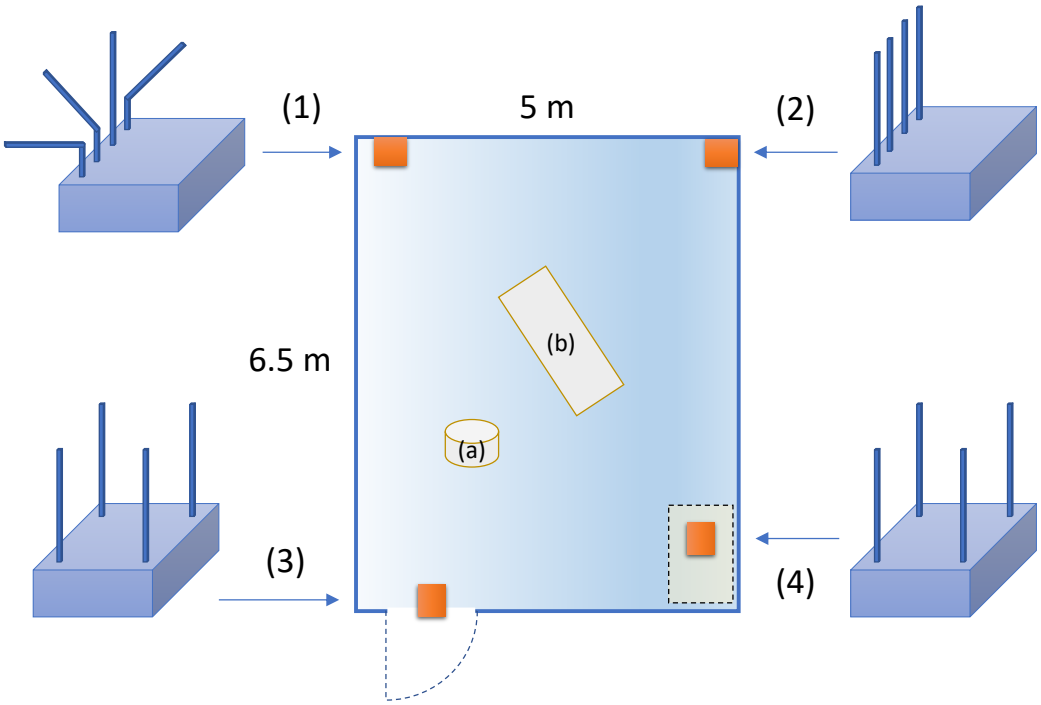
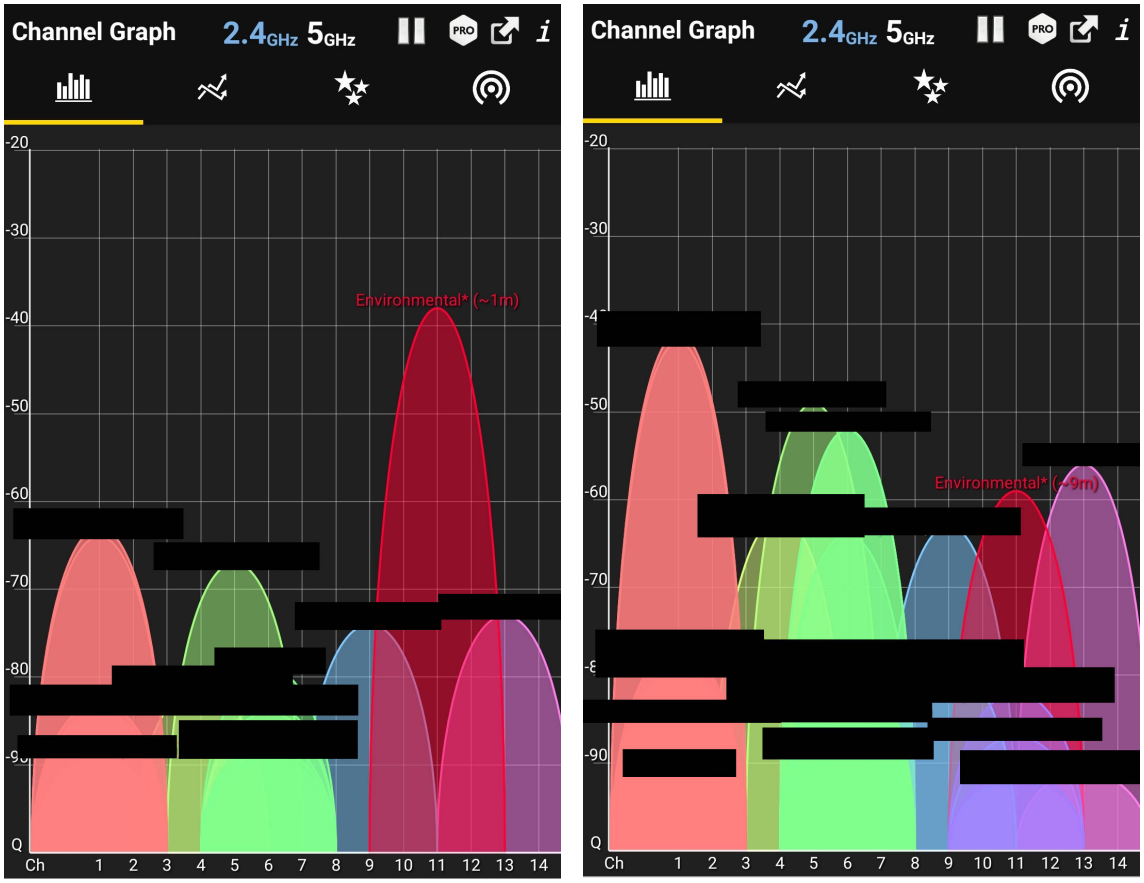


Figure 31: Channel state information collection space set up in Espoo. Room where activities take place and CSI is recorded has dimensions of 6.5x5 m. Wi-Fi CSI acquisition devices (1-4) are placed in corners of the room for the maximum fan-out. (1-3) are STAs, while (4) is AP. The latter is hosted within a PEAD device and displaced in between CSI recordings – every several minutes. STA (3) is placed above the front door. Antennas of (2) are arranged into linear array, while antennas of (3) are arranged into a strange pattern to increase variability of recorded CSI data. (a) is a chair to sitting down / standing up from, while (b) is a yoga mat (and later a proper bed) for lying down and getting up from.



Inside Outside

Figure 32: Wi-Fi signal strength measured inside and outside the data collection chamber with metal-layered walls (depicted in Figure 31). The *"Environmental"* is the name of Wi-Fi network from the router within the chamber (red parabola). All other networks in the images are from outside sources. The signal drop is estimated to be around 20 dB (power) and is happening presumably due to metal layered walls of the data collection chamber. Note that measurements are performed for 2.45 GHz networks, while CSI measurement network actually operates at 5 GHz. Such inconsistency is due to the author's exacerbation of chronic intellectual deficiency. On the other hand, one may expect the received signal strength drop for 5 GHz to be even more pronounced due to the skin depth being inversely proportional to the square root of frequency [Cheng, 2014].

Figure 33: Panoramic views of the "OpenOffice" data collection space from different angles. In contrast to the first collection space "Merlin", the second place features wide areas with different potential Wi-Fi radiation propagation paths.

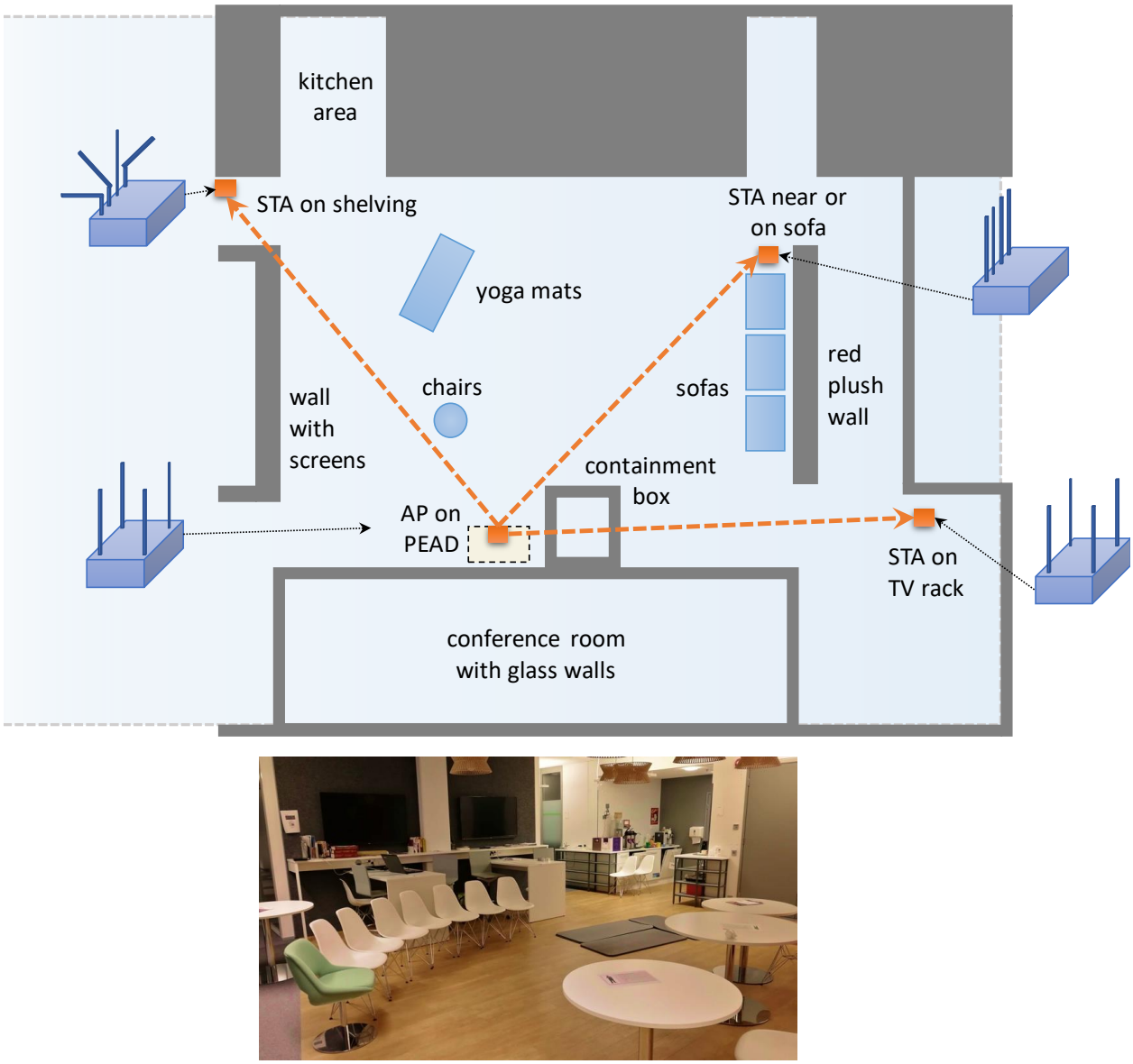


Figure 34: Upper schematic: The plan of the second data collection space code-named "OpenOffice". Only about a third of the whole space is included in the plan while the remaining part is left out of the mapped area. Lower image: in order to increase the diversity of the collected CSI activity patterns, chairs and yoga mats have been placed in specific orders before the recordings. For instance, arranging chairs in array allowed to stand from one chair and sit on another, covering the whole array throughout the experiment. As for yoga mats, lying several mats in a rectangular formation allowed to lie on them from any direction.

Dataset	Number of samples	% from 1 person
Sunnyvale	4608	~10%
Shanghai	680	100%
Espoo Merlin	4400	~82%
Espoo OpenOffice	5400	100%

Table 1: Datasets available for activity classification network training. Dataset collected in Shanghai appears to contain the least amount of samples. This might be due to the fact that majority of samples from Espoo and Sunnyvale have been recorded with 80 MHz bandwidth, while samples from Shanghai with 20 MHz. During raw data preprocessing phase, one 80 MHz sample has been sliced into four 20 MHz samples to ensure a unified shape for NN input (see Figure 37). Espoo datasets have been recorded by the present work. In Espoo Merlin dataset, 3600 samples are from one person, while 800 samples are recorded from six other people. In Sunnyvale dataset 4544 samples are recorded for one person, while other 64 are equally shared by two other people. Shanghai dataset is fully collected from one person, while Espoo OpenOffice from another. Resulting NN prediction accuracy varies greatly depending on training and testing datasets selection.

Shape of tensor prior to top layers: (9x7x8)			
Top layers, *Case 1*		Top layers, *Case 2*	
layer	*parameters*	*layer*	*parameters*
Dense(900)	454500	Dense(90)	45450
Dense(180)	162180	Dense(6)	546
Dense(6)	1086		
Total	617766		45996

Table 2: Fully-connected top layers stacks number of parameters comparison.

for the everything-else-agnostic neural network. In a localization case, however, one has to take into account a particular environment configuration and cannot realistically expect environments to share a similarity to the extent human bodies do. For example, apartments can possess different number of rooms and corridors, vary in ceiling heights and be differently furnished. These unknown variables necessitate fitting or even re-training a neural network to a particular environment. Consequently, one of the virtues for localization NN would be swift and inexpensive training. One straightforward way to achieve it is to decrease the number of trainable parameters.

In the course of the present work the localization architecture has been re-created to incorporate ResNet and Inception features described in Section 2.4.1. The minimal form of a resulting architecture with two parallel branches with two residual blocks per branch is described in Figure 35. The novel network achieves decrease in number of parameters from ~58.8 to ~0.4 millions by using just two methods:

- Utilization of a (1x1) point-wise convolution layer in order to compress the number of channels in the pivot point. This pivot point is selected since:
  - It is an input to the fully-connected layer. Since the dense layer number of parameters is $(inputs) \times (outputs) + (outputs)$, squeezing the number of input channels decreases the number of parameters proportionally.
  - At this point the concatenation of parallel branches outputs takes place. This channels-wise concatenation obviously increases the number of channels by multiple times.
- Decreasing the depth and number of neurons in dense layers stacks. Table 2 contains two examples of such stacks. There, Case 2 corresponds to the final localization architecture top layers.

It is worth to remember that in some situations reducing the raw number of parameters does not bring operation speedup. For instance, DenseNet (Section 2.4.1) requires disproportionally large amount of RAM during operation due to the need to store all convolutional layers activation maps within the *dense block*. However, the number of parameters is easy to evaluate and it serves as a helpful rule of thumb

during network optimization. Further methods to reduce the localization neural network size are described in Section 5.3.1 of Conclusion and discussion.

In one of the prior data collection locations, two parallel independent Wi-Fi networks were installed. In other words, one AP has been connected to three STAs and another AP has been connected to other three STAs. Localization neural network trained on data collected with one Wi-Fi network could not effectively predict the human body location from samples collected with another Wi-Fi network. This observation further solidifies the presumption that the localization network should be re-trained for every particular environment and AP and STAs positions.

4.4 Environment agnostic activity classification

This section starts with description of an implemented pipeline for slicing and first-stage preprocessing of human activity CSI recordings. It continues with adding a differential phase layer to previously amplitude-only samples in Section 4.4.2. Section 4.4.3 describes the abandoned early groundwork for FFT-preprocessed amplitude-only short samples classification. The Section continues with environment agnosting data augmentation methods in Section 4.4.4. The following 4.4.5 describes a NN architecture with novel features dedicated for multi-environment classification. Section 4.4.6 explains an important consideration of an effective receptive field for a convolutional activity classification network. Finally, the Section 4.4.7 presents results from applying data augmentation and agnostic NN features and provides conclusions for future work.

4.4.1 CSI records slicing and interpolation

Raw CSI recordings are between 7 and 20 minutes long. Each contains a series of physical activities similar to the one in Appendix B. The subset for environment agnostic activities classification is "STAND_TO_SIT", "SIT_TO_STAND", "LIE_DOWN" and "GET_UP".

One quickly finds that different activities have different duration. For example, a "STAND_TO_SIT" activity can take less than one second, while getting up from the floor into standing takes full five seconds. In addition, the reaction time to an audio command varies between humans under test. Unlike LSTM networks, which compress past in a single vector [Greff et al., 2017], CNNs do not have such internal memory and need to see an entire activity, or, at least, its defining feature within a sample. Therefore, (1) half a second interval has been reserved before voice command timing and (2) five and a half seconds long activity samples has been sliced for all types of activities (including short ones). This also exonerated data augmentation procedure from adding padding for NN input data shape unification.

One thing to mention is that due to semi-random delays between CSI frames (see Figure 11 or Appendix C), label borders in audio script and CSI measurement do not coincide. For example, label border in the script may be from 15000 ms to 18000 ms, while timestamps in the measurement are: [..., 14998, 15014, ..., 17994, 18025, ...].

Looping over CSI measurement timestamps to find the nearest value proved to

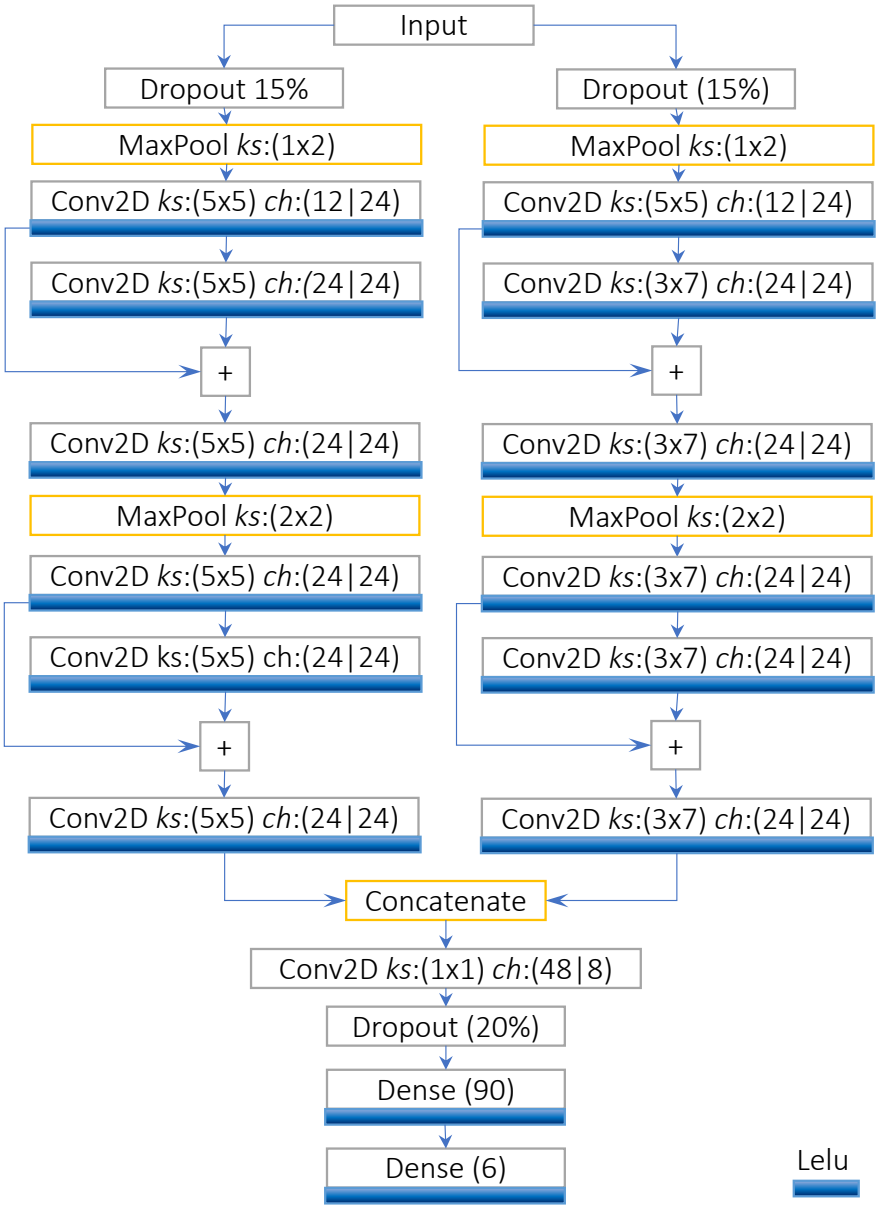


Figure 35: Human body localization NN architecture. *ks* is *'kernel size'*, $ch{:}(x_1|x_2)$ stays for '*channels*:(*input*|*output*)'. The network is rather shallow and contains two branches with slightly different kernel sizes. Each branch includes two ResNet blocks, separated by pooling layer. Prior to fully-connected layers is 1x1 point-wise convolution aimed to decrease the number of first dense layer input dimension by $48/8 = 6$ times. Padding in convolutional layers is $floor(ks/2)$ to keep data width and height constant throughout a ResNet block. Based on this shallow and narrow architecture, a model generator for arbitrary number of branches and residual blocks automated effectiveness testing may be created.

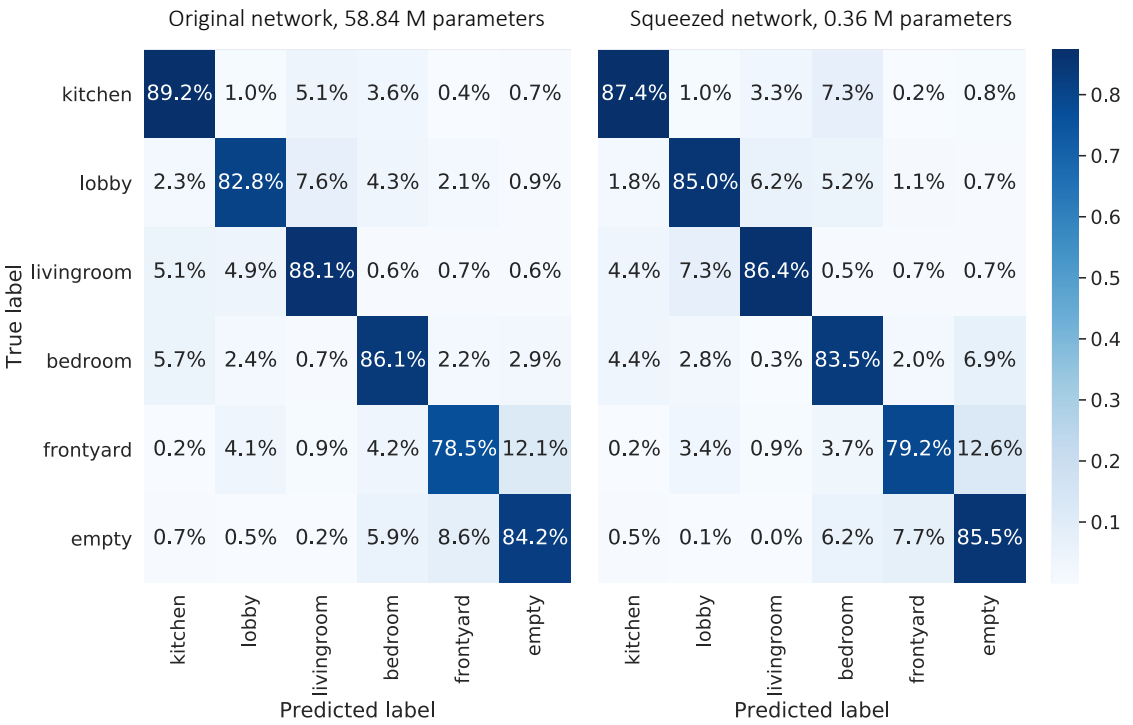


Figure 36: Confusion matrices for human body localization within an apartment. On the left: accuracy for the original architecture. On the right: accuracy for an updated architecture. Overall accuracy drop after network size decrease is 85% to 84%.

be a long endeavour. Fortunately, timestamps are sorted list and it has been possible to use a bisection algorithm with $O(\log(n))$ complexity and locate the closest to the target timestamp faster.

Sliced samples are interpolated to a unified shape, Figure 37. Interpolation period is chosen to be 10 ms – slightly lower than mean (∼18 ms) and median (∼13 ms) for AP-STA links 1-3.

Sliced samples could be divided into train and test sets by the following metadata fields:

1. **Location**. Different physical spaces lead to different Wi-Fi multipath configurations. Network trained on one location should be tested on another to confirm environmental independence (agnostics).

2. **Person**, whose body has been used for activities recording. The assumption here is that different people would have different body configurations and different patterns when performing formally same physical activities. The more people participated in dataset collection and the more even is their share – the more robust NN predictions one can expect.

3. **AP name**. Some locations record data with multiple independent Wi-Fi networks. For example, Shanghai collection place has two separate Wi-Fi networks, each containing an AP and three connected STAs. This allows to collect data from two different perspectives with different multipath configurations. Therefore, in addition to a plain cross-environment, the "same-environment but different viewpoint" testing becomes possible.

4. **Label**. Again, environment agnostic classification has been performed for sitting, unsitting, lying, and unlying activities.

5. **Sample number**. Some samples have been sliced into several during unified shape array formation, as illustrated in Figure 37. In order to ensure that fragments of one original sample do not end up both in training and testing sets, such descendants are tracked by their origin.

4.4.2 Incorporating phase information

One of the goals for the present work has been incorporating phase information as an input to neural network. Therefore, an investigation regarding phase aspect of CSI readings has been conducted. It has been figured out that the CSI phase changes between readings in a seemingly random fashion, as depicted in Figure 39. However, phase offset between different AP antennas or spatial streams, id est MIMO paths, appears to be quite constant.

Three types of phase offsets or deltas have been investigated (Figure 40):

- between links (exemplī grātiā, AP-STA1 and AP-STA2 links),
- between different AP antennas, for example, $(Tx[1], Rx[1,2,3,4])$ and $(Tx[4], Rx[1,2,3,4])$, and

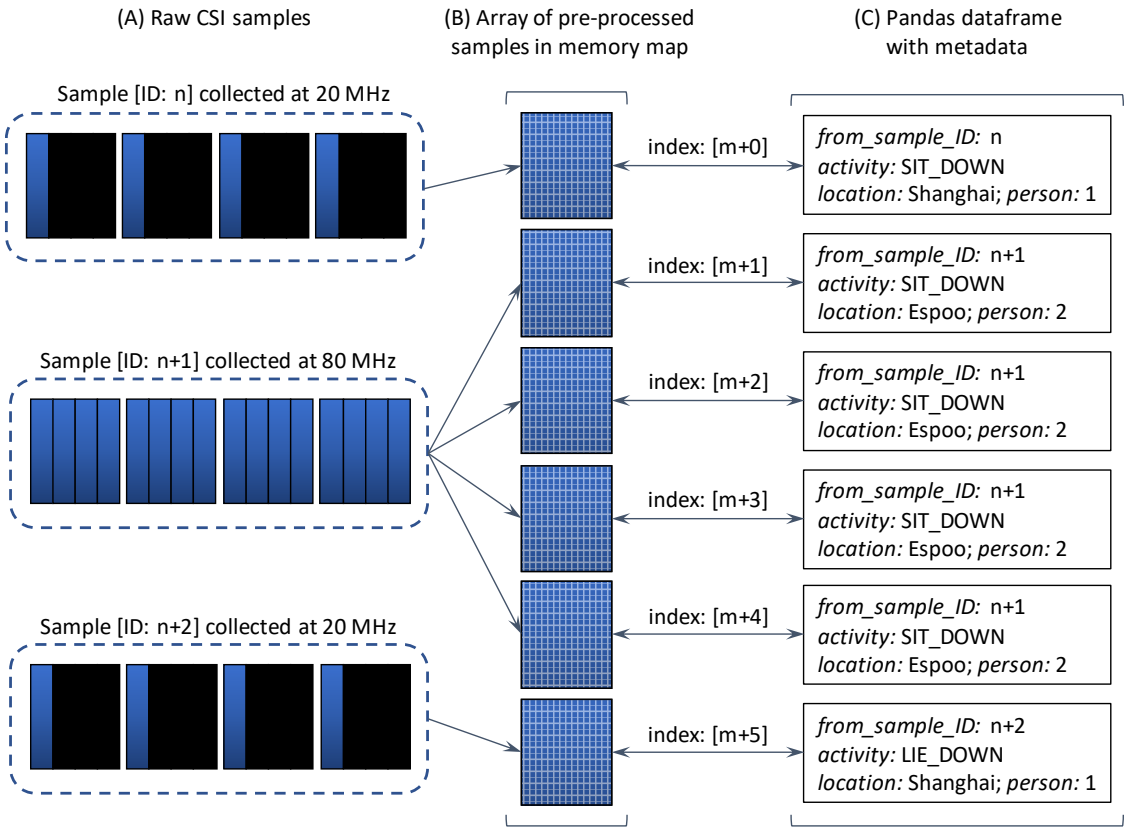


Figure 37: (A): Samples could be sliced out from 20 and 80 MHz CSI recordings. One 20 MHz sample produces one preprocessed sample, while 80 MHz is divided into subsequent ones (B). This is done because array elements have to have a unified shape to be written in a disk memory map. There, reserving 80 MHz space for every sample would result in a disk space waste. In order to distinguish between samples, a separate file with metadata (C) is created. Indexes of metadata collections correspond to indexes of samples in memory map. There, m would be (the number of preceding 20 MHz samples) + 4×(the number of preceding 80 MHz samples). Since the current *numpy* version does not support dynamic reserved memory map file resizing, one has to loop over all samples twice. First, to figure out the number and bandwidth of samples (unloading checked samples from RAM) and reserve the required disk space. Second – to actually preprocess and store the results in the reserved memory map file.

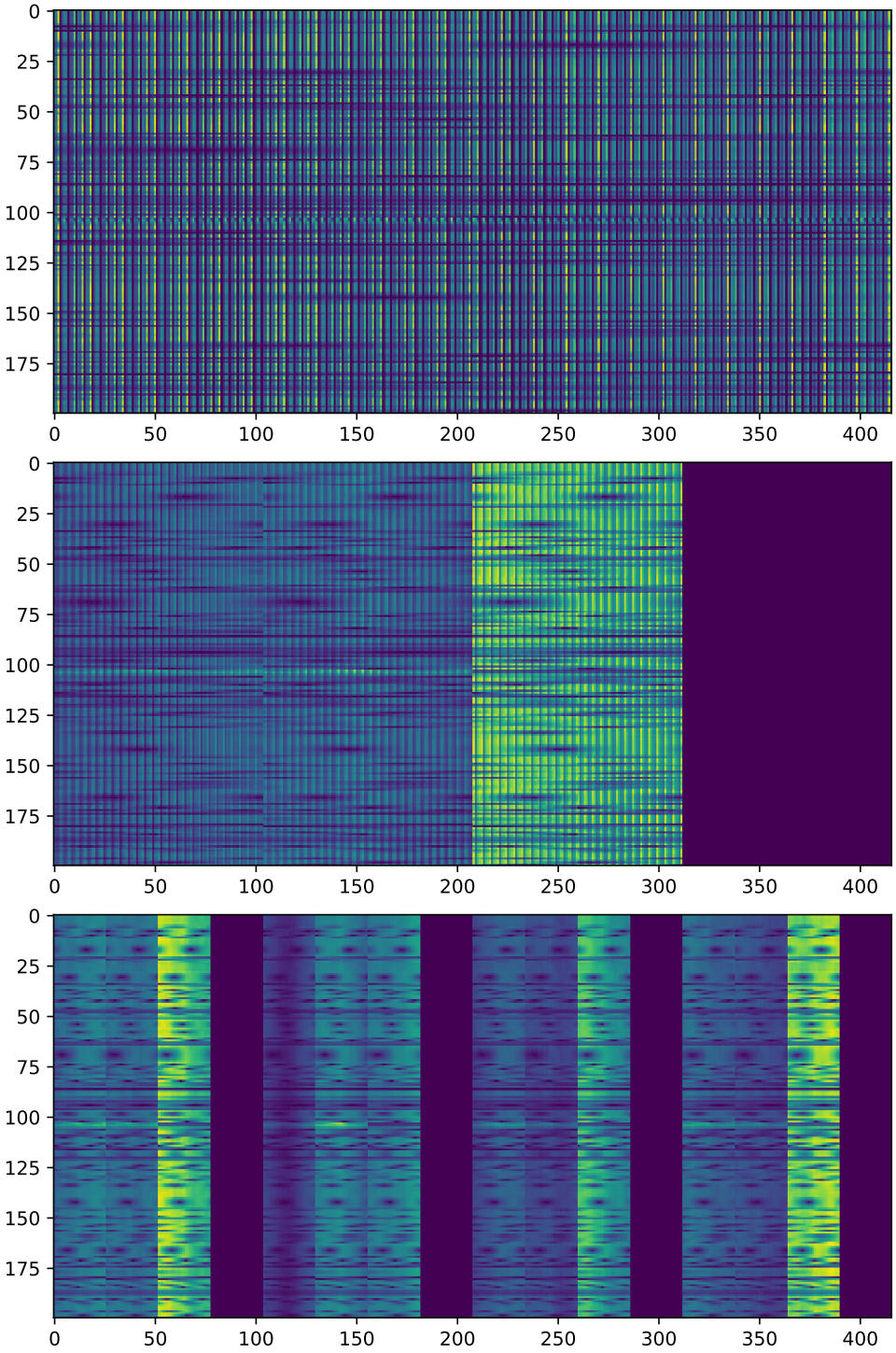


Figure 38: When processing a tensor, it has been found of importance to keep track of tensor dimensions order prior to flattening. Dimensions of all three flattened tensors in the Figure is [*time*, 416]. For the first image, the source tensor had dimensions [*time*, 26, 4, 4] where 26 is the number of subcarriers, 4 is the number of AP antennas and the other 4 is the number of spatial streams. The source tensor for the middle image was reshaped to [*time*, 4, 26, 4] prior to flattening, while the source for the bottom image – to [*time*, 4, 4, 26]. Needless to say that the last variant is preferable for CNN due to the higher correlation between neighbouring pixels and smaller required neuron's ERF (Section 2.4.2) requirement for pattern grasping.

- between neighbour subcarriers in a single (Tx, Rx) MIMO path.

The phase difference *between links* appeared to be the least clear of options. Additionally, one could not guarantee the presence of multiple links in a real life scenario. Comparatively, the number of AP antennas or subcarriers within a CSI report is expected to be strictly more than one. Out of these two remaining options the phase difference *between antennas* has been chosen as a phase information input to a neural network. It has later been found that other researchers in [Wang et al., 2017b] and [Palipana et al., 2018] have made the same choice.

4.4.3 CSI frequency spectrum investigation

During the early work, one attempt to represent CSI samples in a more environment agnostic way was to Fourier transform them along the temporal dimension. The logic behind such transform is that every human movement visible on a CSI sample as waves (Figure 1) and thus has a specific spectrum signature. Such spectrum signature should be less vulnerable to the multipath configuration change (Figure 9) due to combining frequency spectrums from all subcarriers and thus eliminating subcarriers ordering.

In particular, early short 3 seconds samples have been parsed with 0.5 seconds window. This window was Fourier transformed and produced a 1d spectrum. By moving this window the stack of FFT spectrums has been produced. Combined spectrums can be seen in Figure 41 – the resulting stacks are $3 - 0.5 = 2.5$ seconds long. Unfortunately, feeding them to early versions of neural networks lead to decrease in prediction accuracy compared to feeding the original, non-FFT samples. This outcome led to wrapping up further work in this direction by the author. The Fourier transformed samples misbehaviour could be caused by at least three potential reasons:

- CNN can be expressed in terms of Fourier transformation [Vasilache et al., 2015, Lavin and Gray, 2016]. As the reverse is true, CNNs appear to have a native ability to perform Fourier transform on the input data. Therefore, detaching them from the richer original information could explain the drop in prediction accuracy.

- FFTs from each of the 416 subcarriers had been averaged, which led to the decrease of raw training information.

- In the later work the environment-agnostic classifier has been built on phase data. When an input consisted of purely amplitude information, the resulting accuracy in cross-domain testing has been equal to the random guessing. In the times of FFT works, training and testing has been performed on the same domain (data collected in the same environment and then split to train and validation sets). Proper FFT agnostics evaluation could be possible only in cross-domain testing after additional data collection in later stages and for both amplitude- and phase-based FFTs.

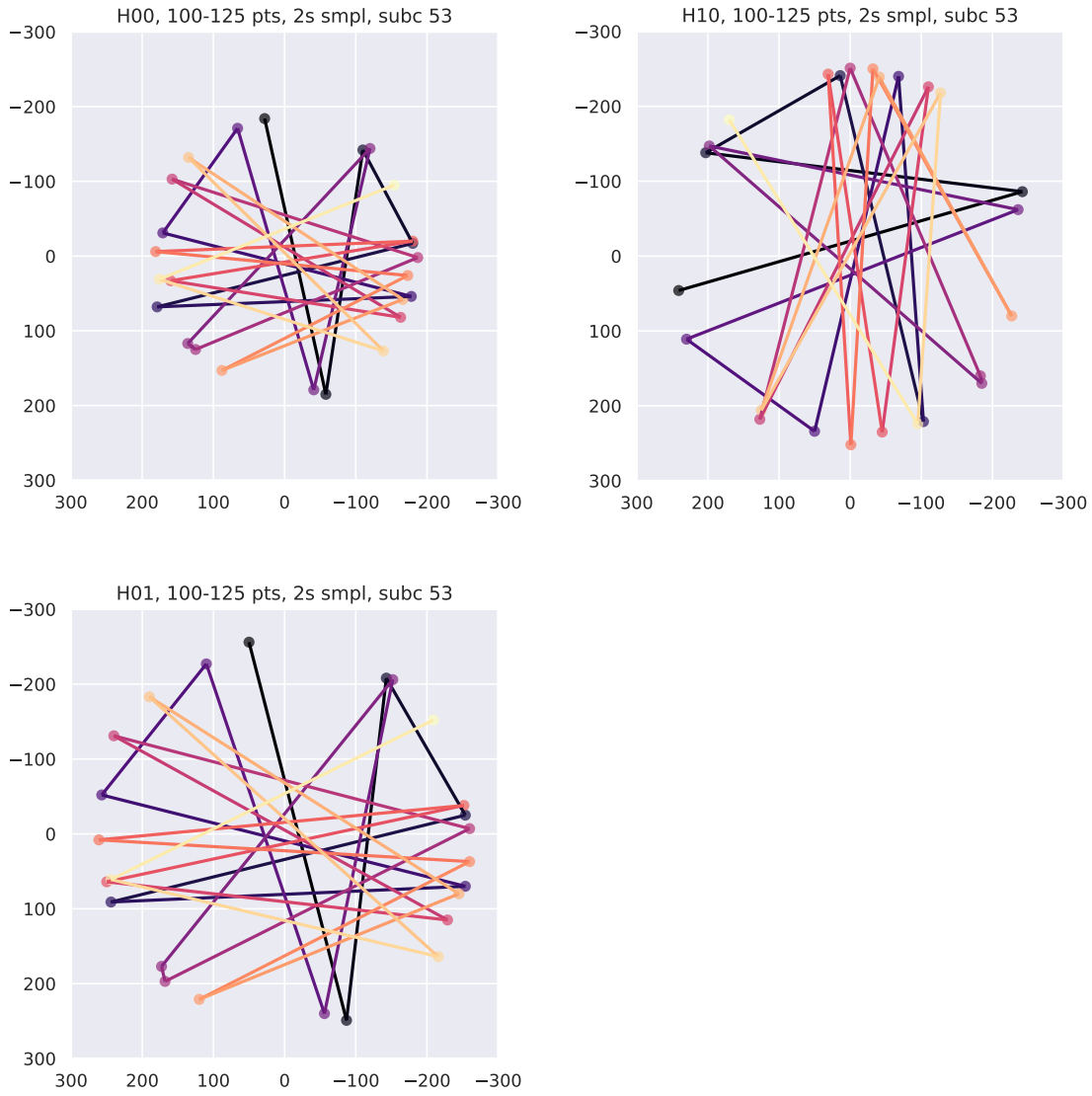
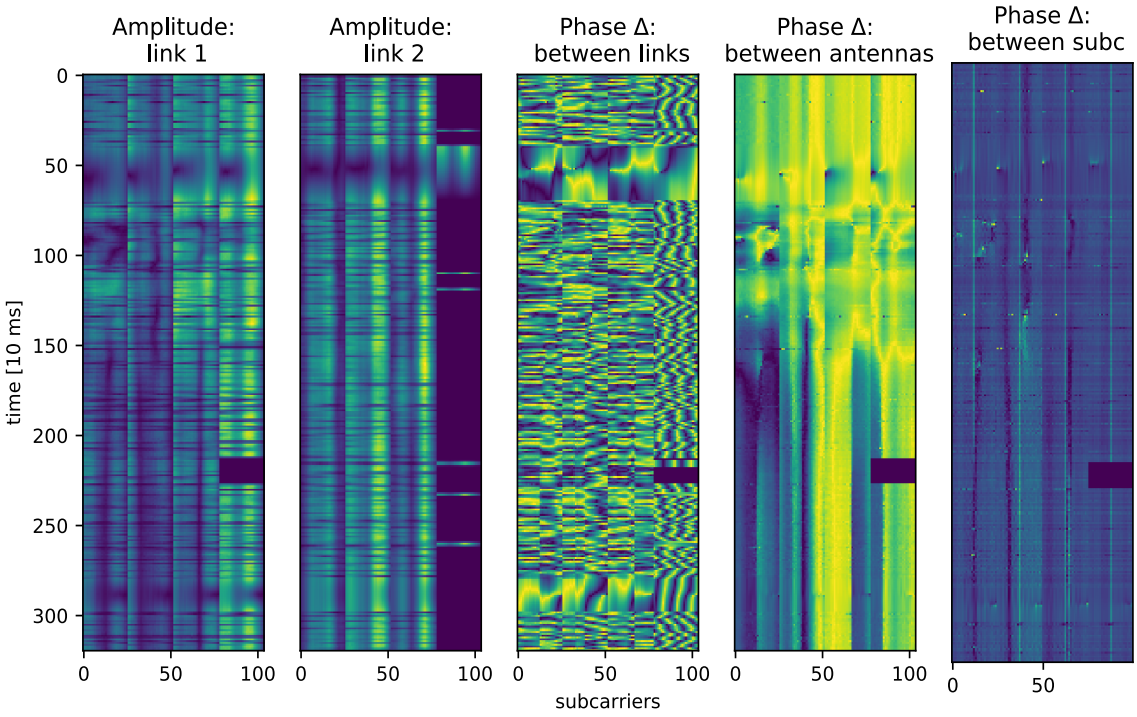


Figure 39: All subplots capture evolution of complex CSI correction coefficient for a chosen subcarrier for 25 CSI reports. Each coefficient value is represented by a dot, subsequent values connected by line; lighter color correspond to earlier readings, darker – to later ones. Vertical axis is for a CSI value imaginary component. Top left: correction coefficient for $(Tx[1], Rx[1])$ MIMO path. Right and bottom: same but for paths $(Tx[2], Rx[1])$ and $(Tx[1], Rx[2])$ respectively. Even though values of CSI coefficient jump all around the circle, it is possible to notice that patterns are scaled and rotated versions of each other.



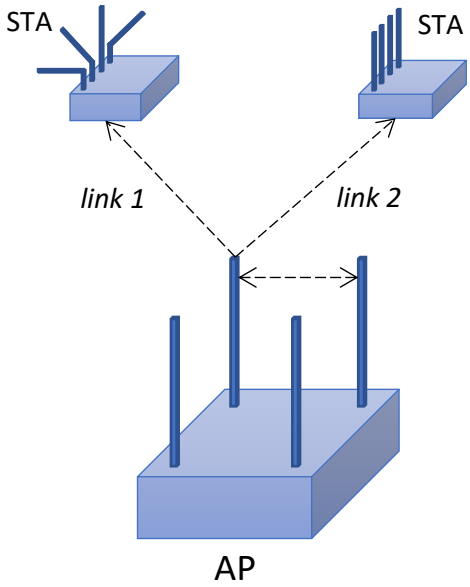


Figure 40: As the previous Figure 39 shows, raw CSI readings phase tends to change in semi-random order. Alternatively, there are at least two ways to obtain a rather consistent differential phase. The first method is by taking phase difference between two AP antennas (different MIMO paths). The second – between neighbor subcarriers within a single MIMO path. These differential phases appear to contain human movement signatures visible with a naked eye. The between-subcarriers option, however, has few high-amplitude outlying readings (bright yellow dots) which rise the necessity to filter the data. Therefore, the differential phase between AP antennas has been chosen for inferencing.

Although the combination of early NN architectures with this form of sample representation has not yielded promising result on the first try, the Fourier-transforming of samples prior to prediction still might be crucial in designing lightweight activity classifiers due to condensing the bulky CSI stream of 416 subcarriers to a much leaner spectrum representation. The supporting factor to this viewpoint could be a paper utilizing FFT on samples to train a highly accurate fall detector [Palipana et al., 2018].

4.4.4 Data augmentation

As described in Section 2.5.1, one of the possible reasons for neural network prediction capability degradation after Wi-Fi CSI acquisition device displacement is the redistribution of subcarriers correction coefficients amplitudes (see Figure 9). After CSI is collected, one way to assist neural network in environment-agnostic generalizing is to strip input samples off as much environment-specific information as possible.

First of all, the data is **normalized** per subcarrier. In particular, each subcarrier undergoes mean subtraction and division by its standard deviation. Secondly, data sample is **differentiated** once in temporal dimension. The outcome of these two operations can be seen in Figure 42.

Aside the environment agnosting alterations, the set of rather standard augmentations aimed to slightly increase the diversity of training data is applied. Whole samples are **mirrored** with 50% chance. Additionally, spatial streams within the sample are mirrored with the same probability. For example, if the data in the original sample was AaBbCcDd, after the whole sample mirroring it would be dDcCbBaA and after subsequent spatial streams mirroring – DdCcBbAa.

The **links shuffling** augmentation is supposed to be irrelevant in the final version since NN processes each AP-STA link independently (Figure 45).

A sample can be shifted back in time by up to half-a-second with the **random offset** augmentation. Since the total length of an activity recording is 5.5 seconds, while leaving 0.5 second for a random offset, a sample duration may be decreased to an arbitrary length, for example, from **5 to 3 seconds**.

Finally, the CSI input to a neural network may be chosen to contain **only amplitude**, or **only phase**, or both information layers.

4.4.5 Neural network architecture for activity classification

In order to gain a foothold in solving the problem, a testing of five standard NN vision architectures has been conducted. Architectures included SqueezeNet [Iandola et al., 2016], ShuffleNet [Zhang et al., 2018b], DenseNet [Huang et al., 2017], MobileNetV2 [Sandler et al., 2018], and ResNet18 [He et al., 2016].

Aside from the ResNet tested as a baseline, leaner architectures have been chosen in order to speed up architecture tweaking cycles. After the initial training on one domain and testing on another for 10 epochs, the only architecture that has shown marginally better than random guessing results was DenseNet. The final DenseNet-based architecture is documented in Figure 47. Aside from simple hyperparameters tuning such as decreasing the number of *dense blocks*, convolutional layers and

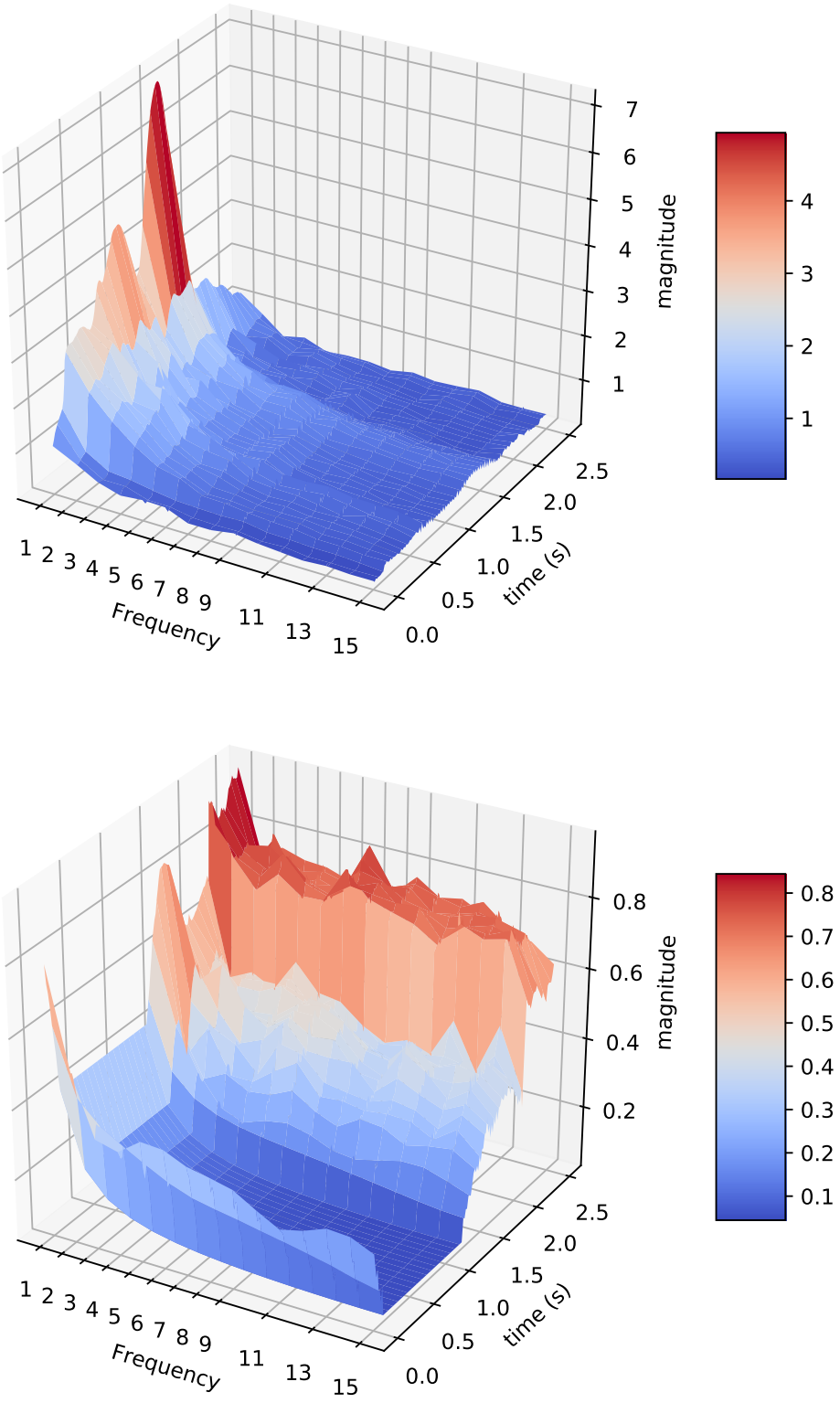


Figure 41: Frequency spectrum of two samples over time, DC component filtered out. Upper plot: no activity is happening. Lower plot: some physical activity is going on.

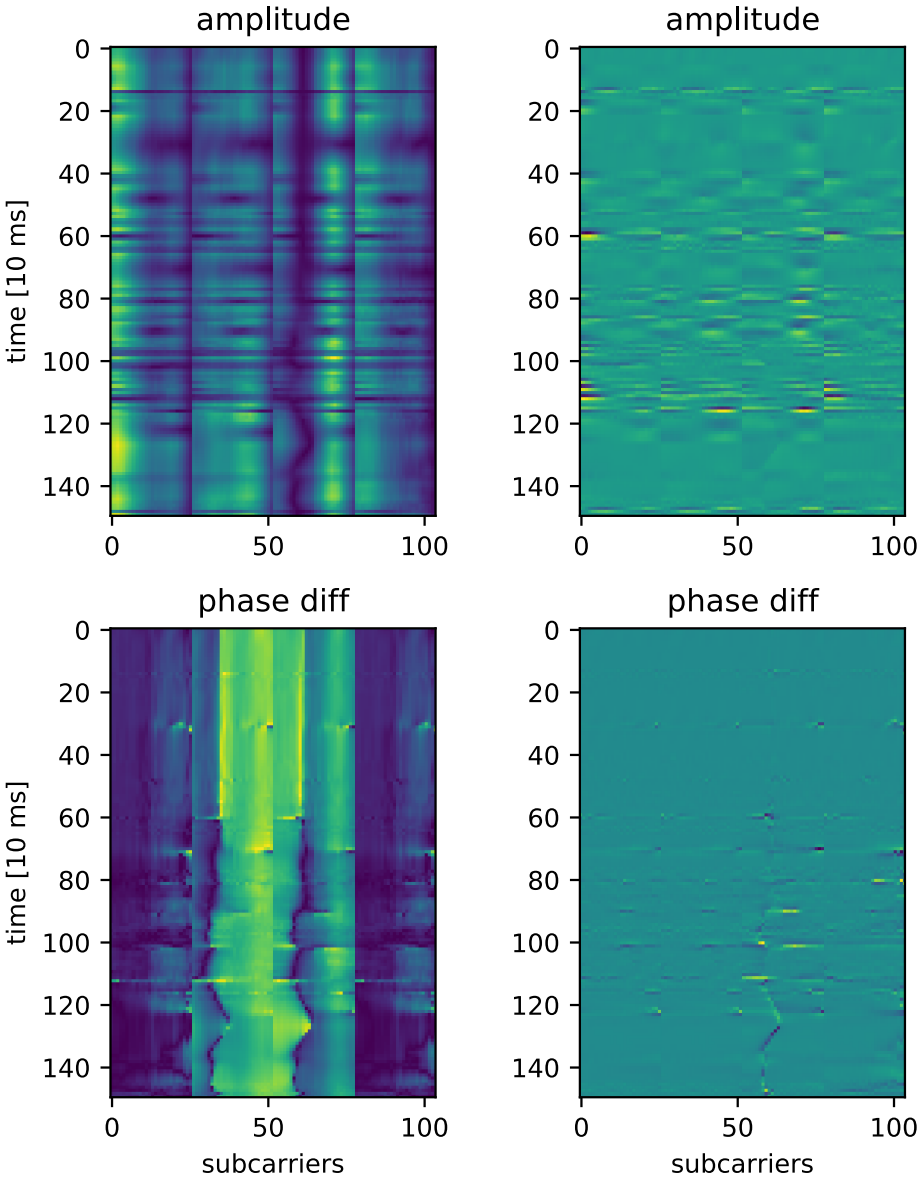


Figure 42: On the left: original amplitude and differential (relative) phase between links. On the right: samples after normalization and first-order differentiation over time. Note that actual samples length is 5 seconds, while plotted is 1.5 s sample part.

number of filters in order to be trainable with the smaller CSI dataset, the resulting architecture features rather novel dense layers which are explained below.

Since the input CSI data differs from a standard image data, it is possible to adapt the network architecture to it. In particular, since each subcarrier in the CSI sample is recorded in parallel, and the physical activity happens at the same time, each of subcarriers could theoretically contain the recognizable signature of such activity. Therefore, it is possible to impose an additional constraint that predictions from subcarriers produced independently per-subcarrier should contain the same answer (activity classification). The method to impose such constraint in the NN architecture is to process a CSI sample with narrow, per-subcarrier kernels and then compare per-subcarrier predictions, punishing the outliers.

Often the loss from the averaged prediction from all subcarriers is significantly less than a mean of losses between each individual subcarrier prediction and the target:

$$mean(\sum_{n=1}^{N} l(x_n, y)) \gg l(x_{mean}, y) \tag{1}$$

where:

- N is the total number of subcarriers, after the samples slicing procedure illustrated in Figure 37 equal to 26,
- $l()$ is the loss function, for example, mean squared error (MSE) or Cross-entropy,
- x_n is the prediction from a single subcarrier,
- x_{mean} is the averaged prediction from all subcarriers, and
- y is the target.

In practice, taking into account the loss between each subcarrier prediction and the target leads to a significant improvement in NN learning speed and achieved accuracy. Note, however, that in order to produce a single answer for a given input sample, averaging of per-subcarrier predictions would still be used.

In order to produce an array of per-subcarrier predictions, either pooling over the temporal dimension (Figure 43) or convolution kernel with height of the whole temporal dimension is utilized. In the latter case, activation map from the lower convolutional layer (I in Figure 44) is convolved by 1d convolutional kernels (II). The height of these kernels is equal to the height of the input (the whole time dimension). Provided this condition, convolutional kernels act as 1d fully-connected or dense layers with shared parameters. Several convolutional layers of (1, whole_height) dimensions can be stacked to emulate multiple sequential dense layers on top of one another. Input goes through the sequence of 1d fully-connected layers and produces one prediction per subcarrier (III) as in a vectorized `FOR` loop. These predictions are compared with a target individually. In case one subcarrier produces some widely incorrect prediction, this prediction is not averaged out and higher overall loss is applied. Another loss component is calculated from the differences between subcarriers pre-last layer activations (IV). The logic here is the following:

since the signature of a physical activity in the sample is synchronized in time between subcarriers, the pre-top layer should produce the same bag of features for all subcarriers despite their potential CSI correction coefficients difference due to the multipath. Here, hopefully, the network would learn to filter out the environment-specific or subcarrier-specific information and focus on general aspects, common for each subcarrier. The total loss is a combination of the target-difference loss and the intermediate-conclusions loss.

Since not only outputs from the last fully-connected layer should be the same (as target), but as well the reasons leading to correct outputs should be alike, it is possible to compare the pre-last layers activations. As the information propagating through the deep NN should lead to generalization – loosing information about particular sample and retaining the higher level, general representation, such approach should be functional.

In addition to between-subcarriers predictions loss and between pre-last layer activations loss, it is possible to incorporate the between-links loss. The logic here is the same as in between-subcarriers situation: since the physical activity is happening at the same time for all active links, the activations of the pre-last layers should be alike. Therefore, they are also compared and the appropriate loss is issued, Figure 45.

4.4.6 Effective receptive field tests

Smallest relevant features in time-differentiated between-antennas phase layer (Figure 42) may be few pixels in time dimension. Therefore, dilation should not be used as a method to increase the effective receptive field (ERF, Section 2.4.2) for at least few input convolutional layers.

A non-recurrent neural network sees a sample of particular temporal duration as a single-piece tensor (with time being just one of the dimensions). The reasoning behind increasing ERF in time dimension is as follows. In order to recognize an activity from the sequence of features, neurons of upper layers should have an ERF comparable to the activity duration. In turn, longer activities such as "standing up from the floor" could take whole temporal length of a sample (five seconds or 500 interpolated CSI reports).

Provided the above logic, several models with different effective receptive field sizes summarised in Table 3 have been tested. The first model had the largest ERF, easily overhanging the 500 pixels long sample.

The test results, however, quickly revealed that shrinking the ERF (*Intermediate model* in Table 3) does not cause dramatic prediction results degradation up until the *Final model* ERF size. The latter has been left to persist as smaller convolutional kernel sizes had a benefit of faster model training.

4.4.7 Environment agnostic ML results and conclusions

On the machine learning part, the agnostics of a neural network towards a particular rooms and obstacles configuration is realized in two aspects. On the one hand, the

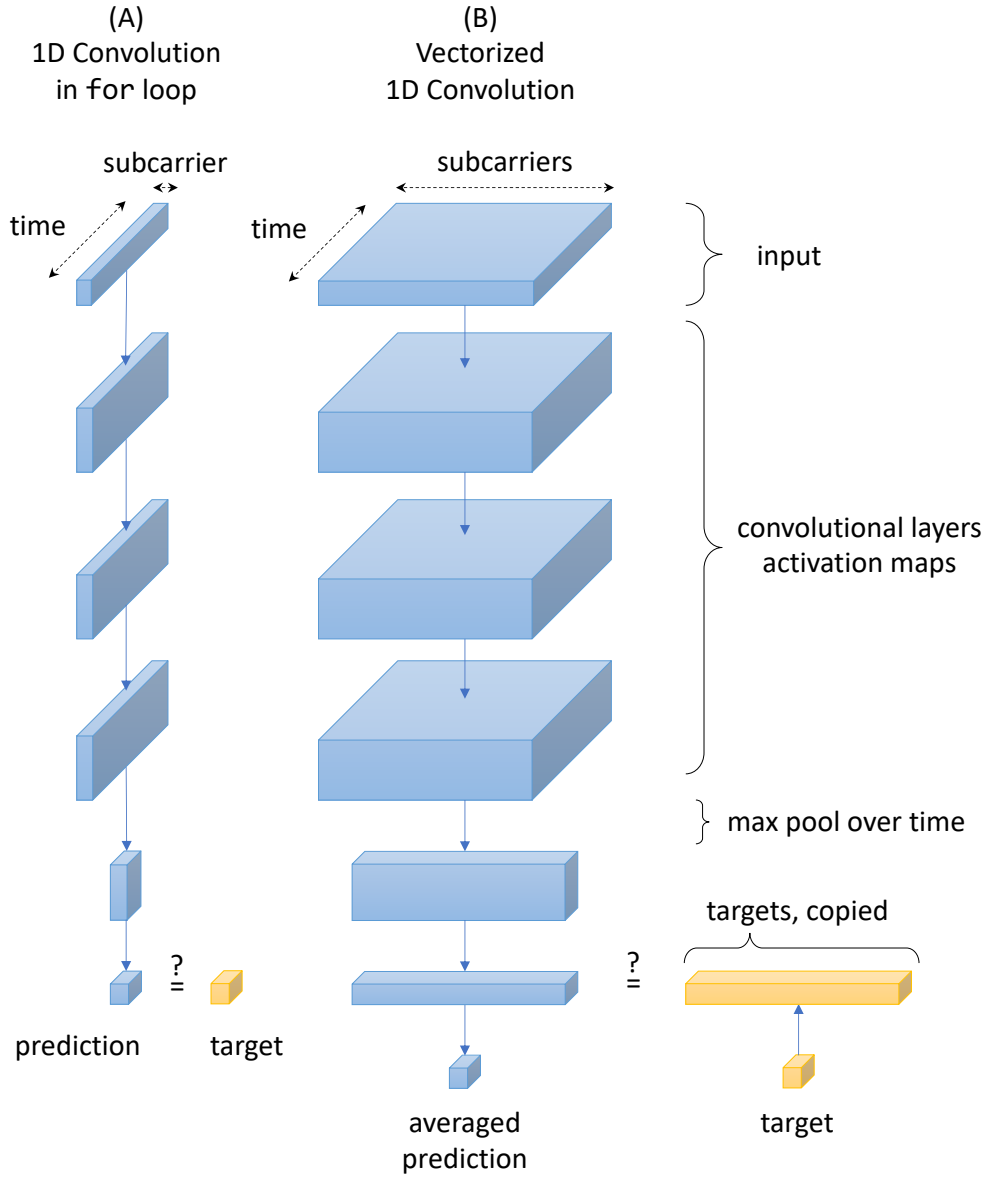


Figure 43: The idea behind per-subcarrier predicting is that each of 104 signal subcarriers could bear a recognizable signature of the ongoing activity. Then a narrow network making per-subcarrier predictions and then averaging them could be trained (A). I.e. this is equivalent to making predictions per-subcarrier in a FOR loop and then averaging them. This process is vectorized in (B). Kernels are kept narrow to emulate the 1d case and their parameters are automatically shared. Produced per-subcarrier predictions are individually compared to target (the $\stackrel{?}{=}$ moments on the plot) and individual losses are averaged. Originally the loss has been produced from a single averaged prediction. However, then "bad" per-subcarriers predictions were averaged out and the network has not been training well. Early versions of activity classification architecture have been collapsing temporal dimension in the pre-last layer with max pooling over whole time, as depicted in this Figure. A more interesting and information preserving approach of designing 1d time-wise fully-connected kernels is illustrated in the following Figure 44.

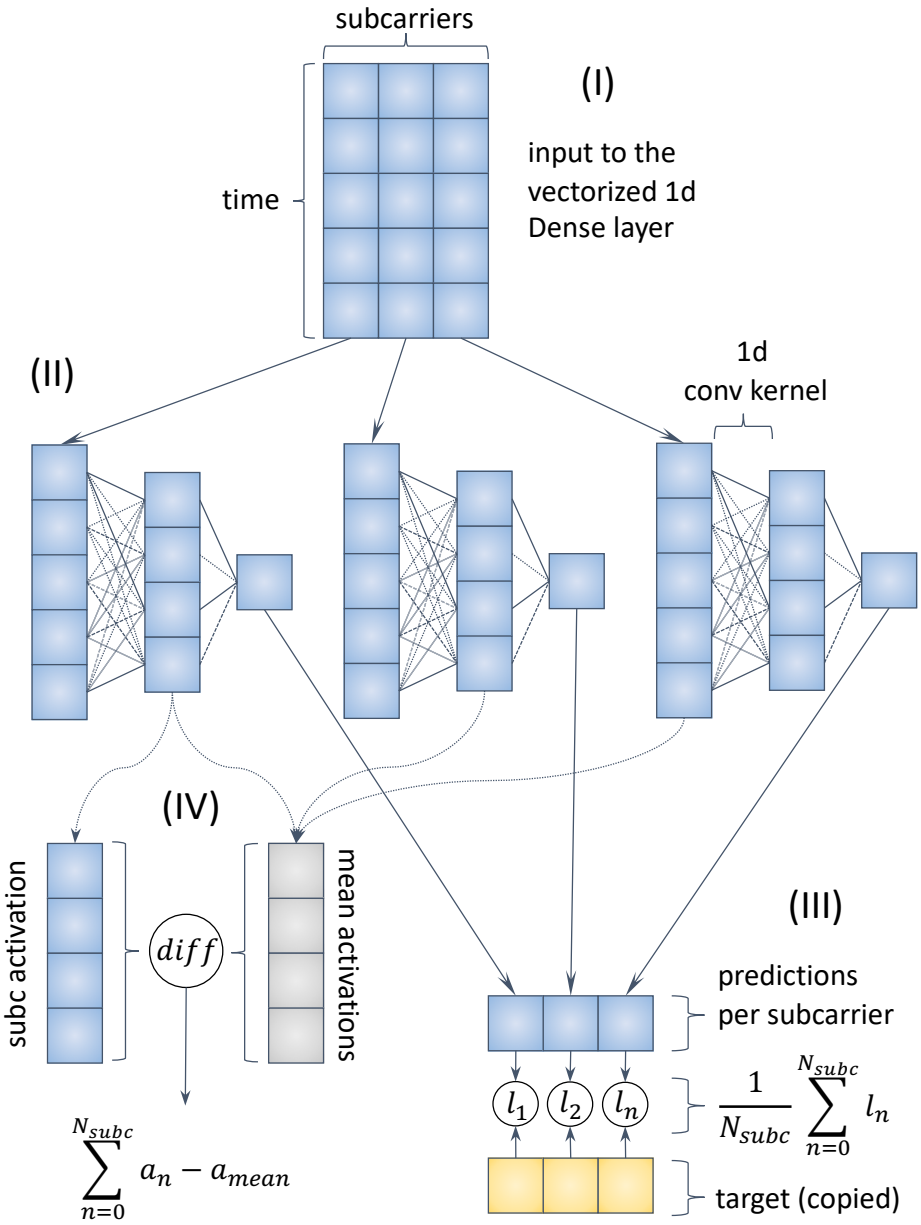


Figure 44: Top "dense" layers of activity classification network and the applied losses. 1d fully-connected per-subcarriers layers are emulated by narrow convolutional kernels. The loss between intermediate "dense" layers is applied. The loss between prediction and target is cumulative and takes into account the difference between each per-subcarrier prediction and the target result.

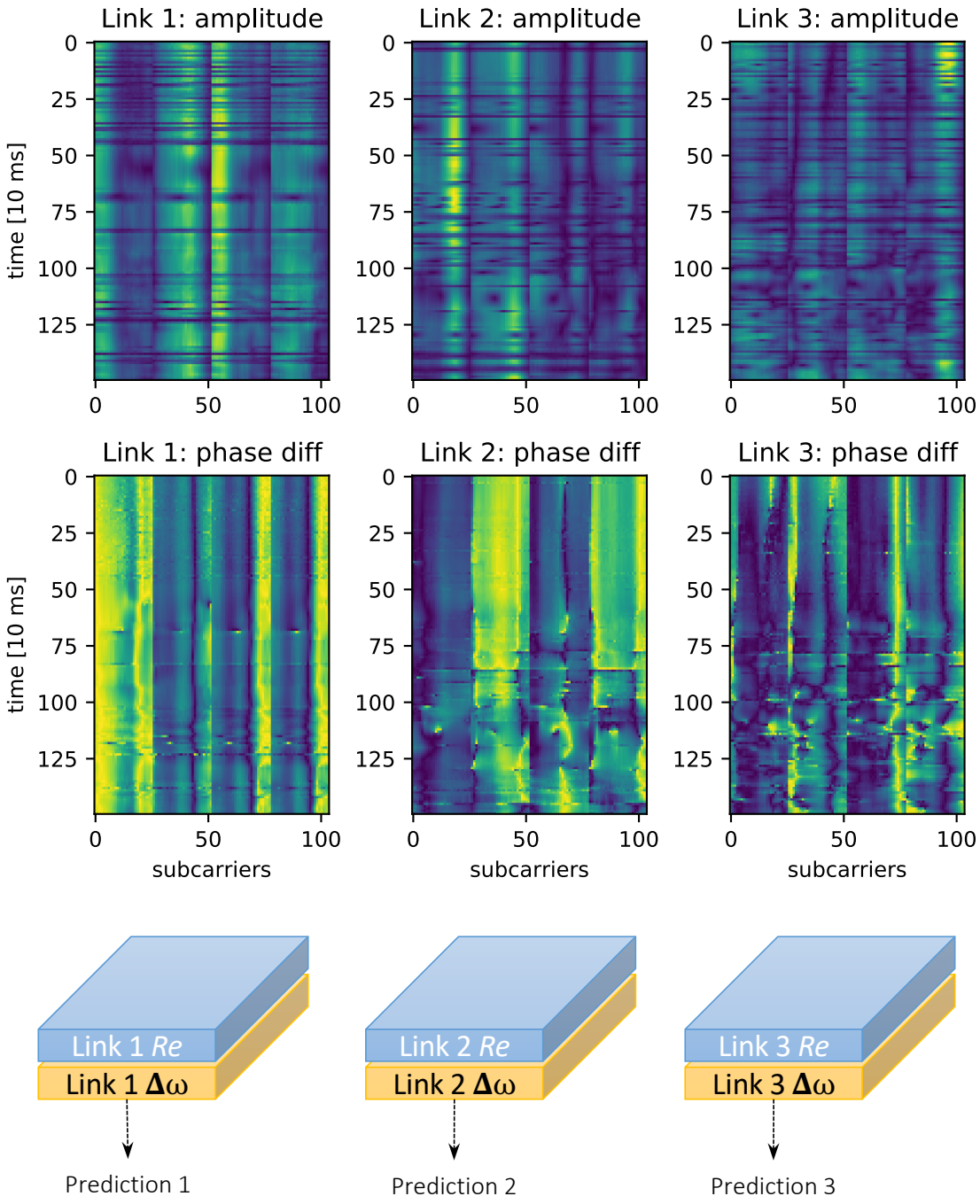


Figure 45: In the practical scenario the number of active links is variable and could be less or more than three. Therefore, in later iterations, NN accepts phase and amplitude only from one link at a time. Separately produced predictions from AP-STA links are averaged. Figure 44 explains how activations of pre-last layers of 1d fully-connected kernels are compared and additional loss is applied if said activations differ. Similarly, the intermediate layers activations are compared between separate links and, if they differ, extra loss is applied. The reasoning behind such similarity constraint is that since NN should be environment agnostic, activations of pre-last layer should be the same regardless of where STAs are located.

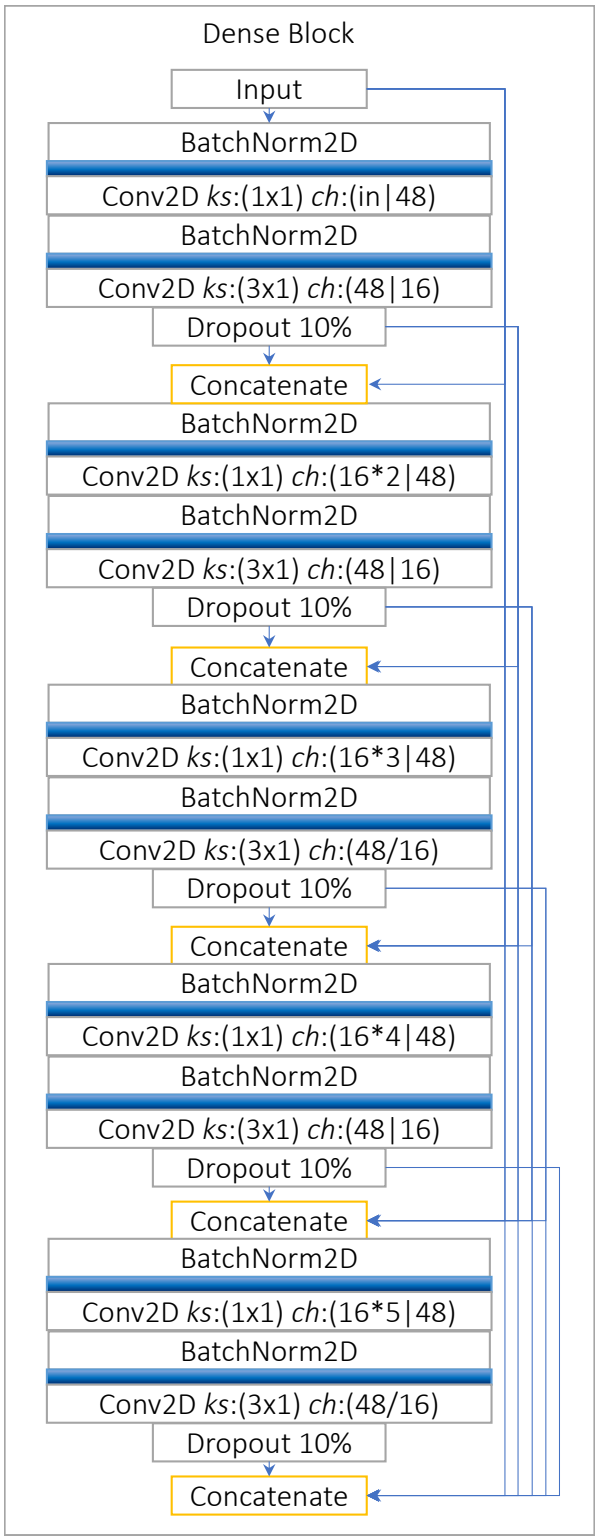


Figure 46: A single *dense block* of a final DenseNet-like architecture described by following Figure 47. The 1^{st} dimension of kernel size *ks* refers to time, while the 2^{nd} – to subcarriers. Convolutional layer channels *ch* are $(input\_channels|output\_channels)$.

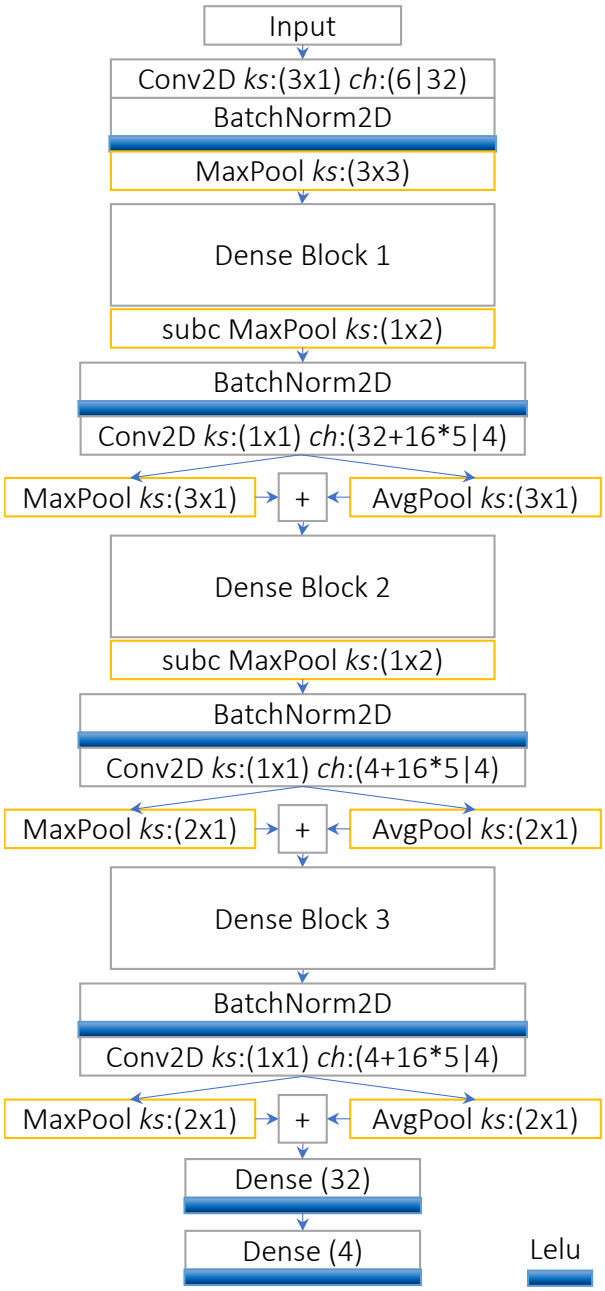


Figure 47: Neural network architecture for activity recognition. It is much smaller than the original DenseNet for ImageNet classification [Huang et al., 2017], and therefore can be trained with much smaller CSI dataset and on a single GPU.

The source used solely average pooling in *transition layers* between *dense blocks*. The present network, however, adds max and average pooling outputs in equal measure; this proportion could be a trainable parameter.

The last two dense layers process each subcarrier separately and are vectorized as 2D convolutional layers as illustrated in Figure 44. For example, kernels dimensionality of the "dense (32)" layer is (*whole_time*, 1) with number of filters equal to 32.

	Final model		*Intermediate*		*Oldest model*	
Layer/Block	effect	ERF	effect	ERF	effect	ERF
At Dense (32)		1		1		1
Max+Avg Pool	×2	2	×2	2	×3	3
Dense Block 3	+10	12	+18	20	+40	43
Max+Avg Pool	×2	24	×2	40	×3	129
Dense Block 2	+10	34	+18	58	+40	169
Max+Avg Pool	×3	102	×3	174	×3	507
Dense Block 1	+10	112	+18	192	+40	547
MaxPool after Input	×3	336	×3	576	×3	1641
Conv2D after Input	+2	338	+4	578	+4	1645
Whole Field		338		578		1645

Table 3: Evolution of effective receptive fields (ERFs) spread in temporal dimension due to adjustments in convolutional kernels sizes and pooling layers coefficients. The *Final model* column summarises the ERF of the model described in Figure 47. Several other configurations have been tested. For instance, in the *Intermediate* model, two out of five convolutional layers of every *dense block* have kernels (5x1) instead of (3x1), resulting in +18 pixels ERF increase per *block* instead of +10. The *Oldest model* had not only all convolutional kernels being (5x1), but also all pooling layers coefficients been raised to 3. As a consequence, it had possessed the largest effective receptive field, easily covering the sample of 500 pixels duration.

input CSI data to a neural network is augmented in order to detach the environment-specific information (Figure 42). On the other hand, the architecture of the neural network is modified to fit the input data and allow for more agnostics (Figures 43 and 44). All results and testing curves discussed in this Section originate from the cross-domain validation. In other words, the network has been trained on one environment and then tested on another (unless otherwise is explicitly noted). Training curves are excluded from this Section plots for clarity.

The comparison of data augmentation methods described in Section 4.4.4 is presented in Figure 48. As it can be seen from testing curves, some of the prepared methods were irrelevant, while few other – were impactful or crucial. In particular, without the normalization to mean and standard deviation, the network could not learn to perform any relevant activity classification until epoch 15 and the final plateau results have been significantly lower compared to the reference. The length of input CSI samples has proved to be of utmost importance in classifying target *sitting down, unsitting, lying down, and unlying* activities. Therefore, the hypothesis that a network should be exposed to the whole duration of an activity has received a supporting evidence. Relevant to a lesser extent have been found the second specific environment agnostic augmentation of time-wise first order differentiation as well as the generic mirroring augmentation. Adding random offset up to 0.5 seconds along with links shuffling augmentations appear to be irrelevant. The latter is not unexpected since, as described in Section 4.4.4 and illustrated in Figure 45, final versions of agnostic NN process each link independently and separately, thus rendering the order of links inconsequential.

Aside from augmentation methods, it has been found that without *(differential)* phase, the network could not be trained on a bare *(differential)* amplitude. Moreover, excluding CSI amplitude layer from the input has demonstrated a substantially higher training speed and testing classification accuracy during the first few epochs. On the plateau stage, however, testing accuracies converged, further solidifying the evidence that either the differential amplitude does not contain relevant information or the final architecture is unsuitable to make a use of it.

In order to verify the irrelevance of the remaining augmentation methods, the long-term (or many-epochs) test has been conducted. Without differential amplitude, random offset nor links shuffle, the test accuracy on plateau stage has been comparable yet consistently slightly smaller (87.2%, std 0.75% vs. the reference 88.6%, std 0.49% in the last five epochs). The difference could be attributed to the absence of the random offset augmentation intended to be beneficial during the later stages of network training for overfitting prevention.

Shifting the topic towards the multi-environment or agnostic NN architecture described in Section 4.4.5, its effect can be evaluated from Figure 50. Although less pronounced compared to the effect of data augmentation (Figure 48), switching on narrow per-subcarrier kernels and applying between-subcarriers and between-links losses for otherwise the same DenseNet-based architecture results in a solid 7% difference in test accuracy. The accuracy increase appears to continue in later epochs, which indicate the significance of the new regularizing losses after the initial training phase. This opens an opportunity to test a dynamic losses coefficients increase at

plateau stages.

The resulting accuracy of an activity classifier greatly depends on the combination of train and test domains, Figure 51. There, data collected from similar humans appears to yield better results. In order to verify this claim, additional testing summarised in Figure 52 has been conducted. There, a NN classifier trained on dataset collected on one person in one location is tested on dataset collected on several people in another location. Neural network appears to be environment agnostic yet human-dependent. Such bonding may be attributed to physically varying proportions of human bodies parts [Daniels, 1952] as well as to individual variations in movement patterns. This rises the necessity to incorporate as diverse human data in the training dataset as possible. The diversity could be increased in at least the following ways:

1. Direct increase of the number of data collection participants. Although simple way with highest fidelity it is costly and organizationally demanding.

2. Instructing participants to intentionally vary movement patterns within for an activity. For instance, sitting down in three different ways has been utilized during OpenOffice dataset collection. Removing shoes for some of CSI recordings may introduce noticeable difference in sitting and lying activities.

3. Induce changes in movement patterns by applying drugs such as caffeine or sleeping pills inward the test subject.[3] Performing data collection during different times of the day and night is a variation of this technique.

4. After dataset collection the differences in CSI samples collected from different humans can be analyzed and "intermediate" samples generated. The present work avoids the simple time "stretching" or "compressing" form of data augmentation due to fear that the subsequent shift and distortion of the whole frequency spectrum may bring higher frequencies into the focus range of neural network and therefore generate misleading samples. However, upon careful spectrum differences analysis, the human-specific may be altered while other "technical" ones kept intact. Therefore, a use of techniques similar to the Phase Vocoder introduced by Flanagan and Golden [1966] may be investigated for the needs of synthetic CSI data generation.

The primary reason for aggregated accuracy decrease is a severe misclassification of a single activity category (top confusion matrix in Figure 53). Often the confused category changes back and forth in subsequent epochs while preserving the total accuracy intact. This differential misclassification is characteristic for networks trained on one person and tested on another.

[3] Author has been trying to alter oneself body composition and movement patterns by donating blood during the second day of OpenOffice dataset collection. This method, however, could not be listed as standard.

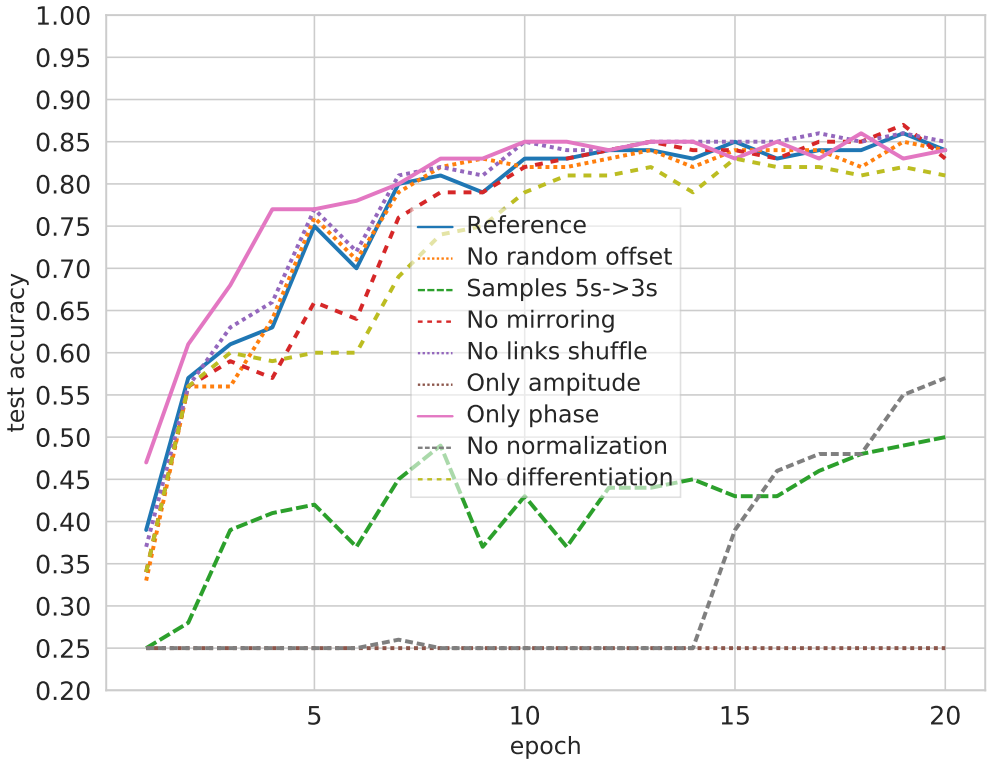


Figure 48: Comparison of data augmentation methods by switching them off one at a time. The most relevant augmentations switch off results in lower curves. Without the per-subcarrier normalization, NN started to train later and the prediction accuracy plateau did not exceed 70% at epoch 60 (not shown in the graph).

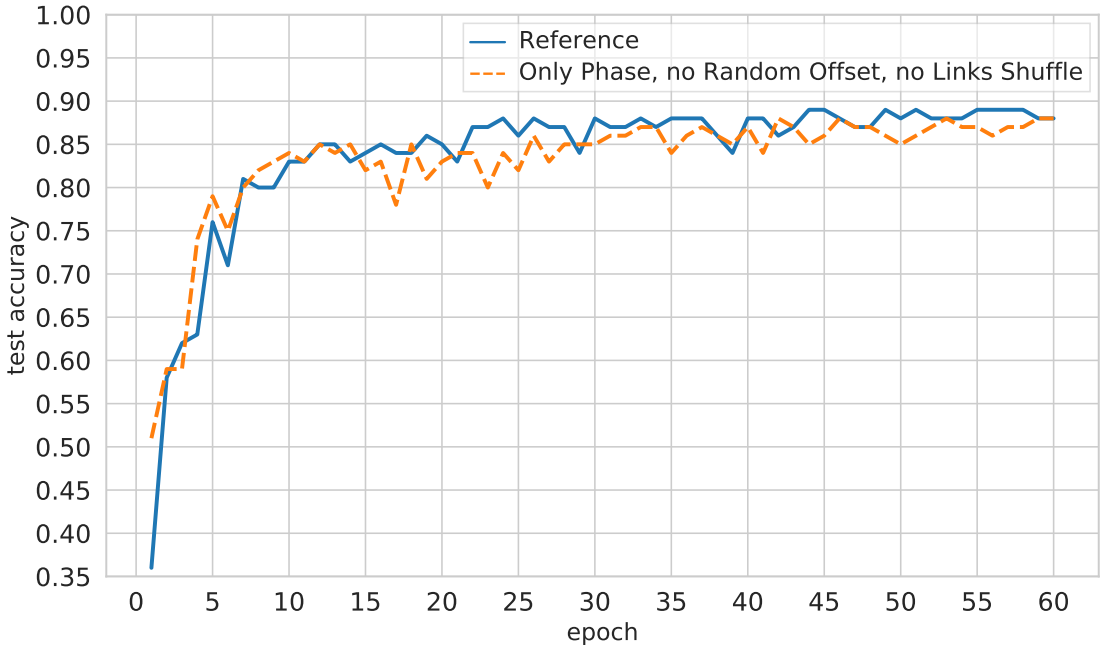


Figure 49: Verification of some data augmentation methods irrelevance. Phase-only samples allow NN to train faster, but the test accuracy during the later epochs is slightly lower, presumably due to the absence of the random offset augmentation.

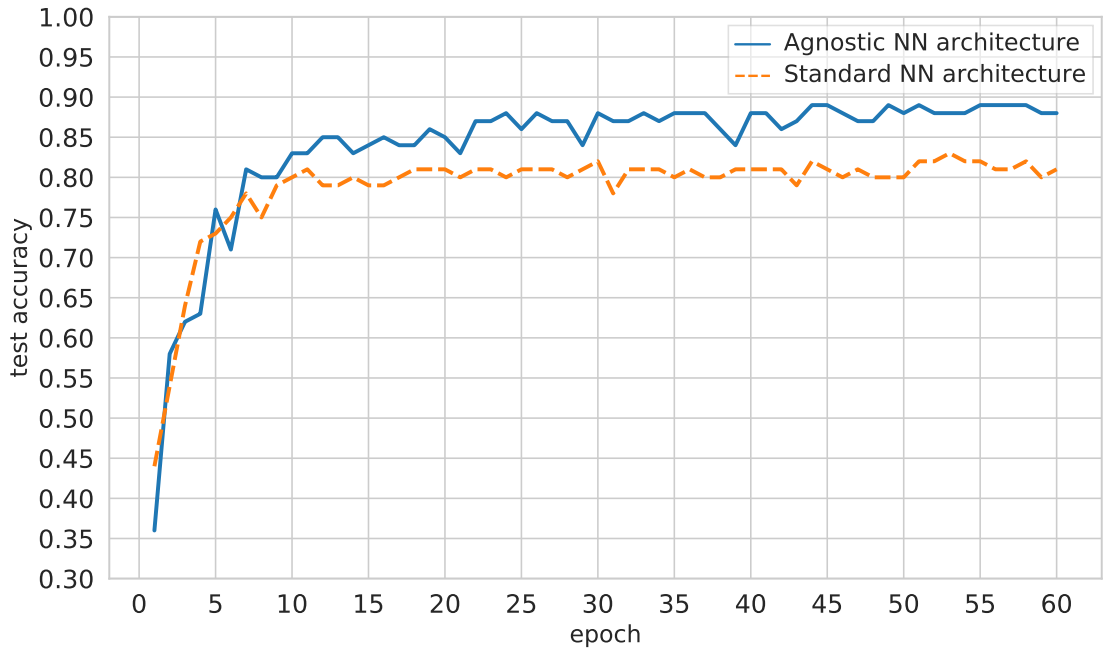


Figure 50: Comparison of the neural network architecture described in Figure 47 with and without environment agnostic features. The dashed curve is NN with standard (3x3) convolution kernels instead of per-subcarrier (3x1) ones, cross-links and cross-subcarriers losses disabled. The average prediction accuracy over last five epochs is 88.5% for the agnostic architecture and 81% for the standard DenseNet-based CNN.

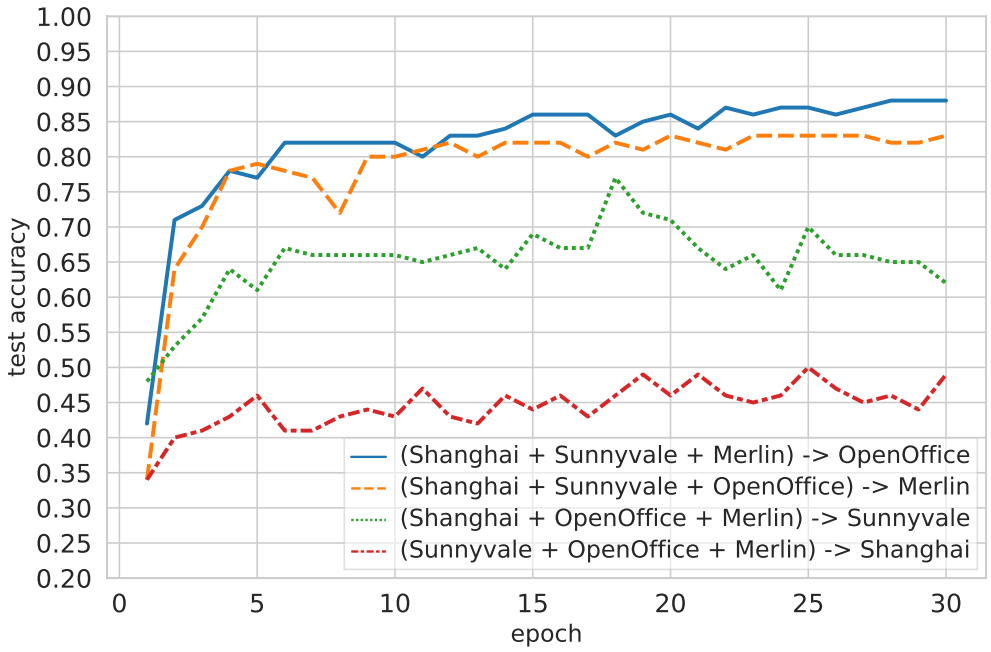


Figure 51: Cross-domain validation. The neural network has been trained on samples collected in three out of four environments and tested on the remaining one. The four domains datasets are described in Table 1. OpenOffice and Merlin datasets collected mostly with the use of same test subject show the best cross-domain validation results. For other combinations it appears that differences in classification accuracy increase with the diversity between train and test subjects. For the validation of this hypothesis, please see the following Figure 52.

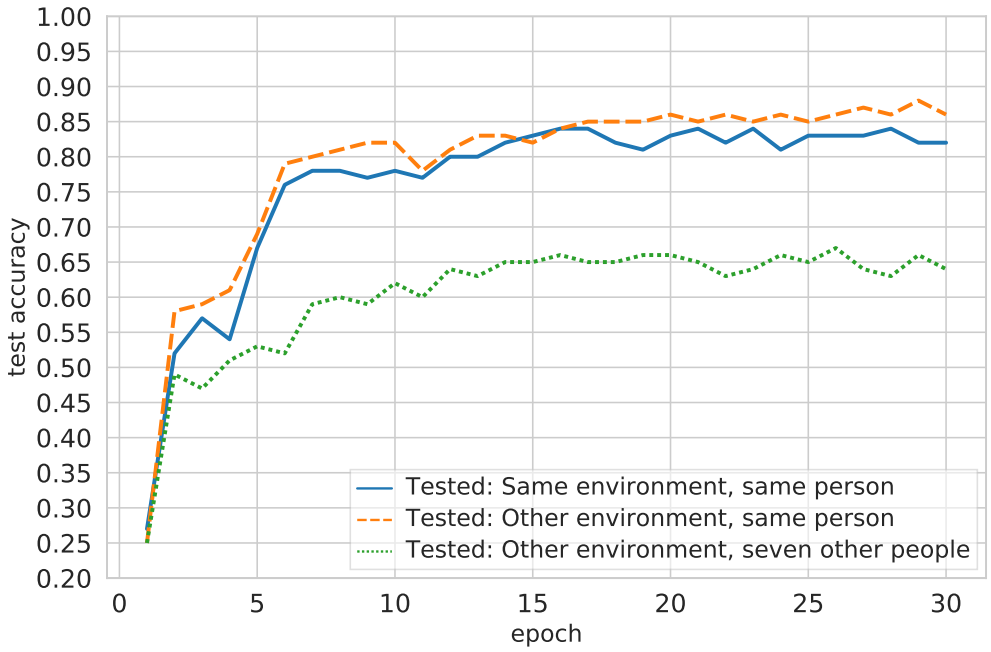


Figure 52: Verification that the difference in cross-domain validation originates from the difference between test subjects. The blue solid curve is the prediction accuracy for NN trained on 80% of OpenOffice dataset (4320 samples) and tested on 20% of it (1080 samples). OpenOffice dataset is collected from a single person. The orange dashed curve if for training on the whole OpenOffice dataset and testing on 3600 samples of Merlin dataset collected from the same person. The green dotted line is for testing on samples collected from six other people in the Merlin environment.

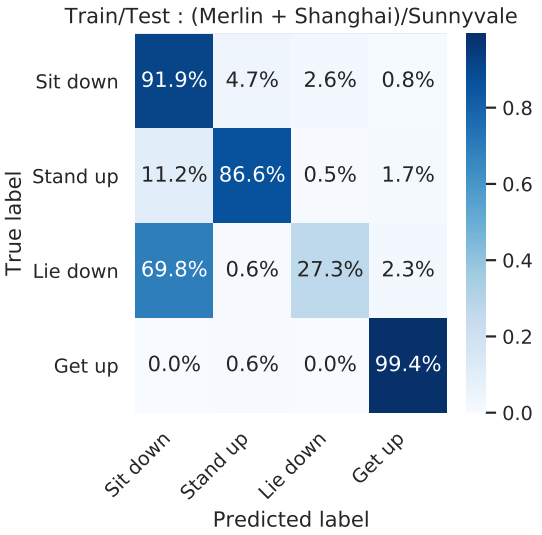
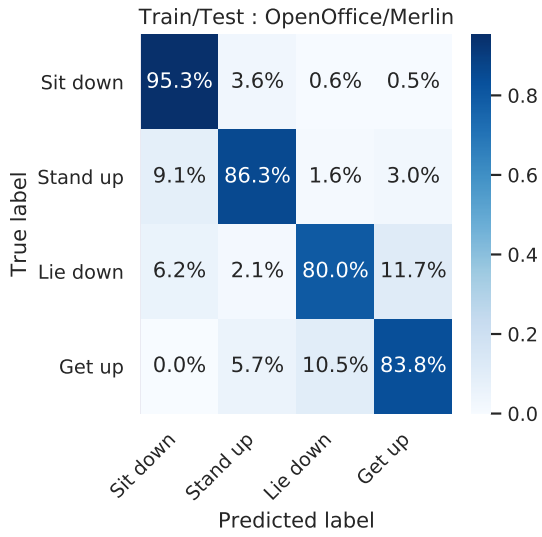
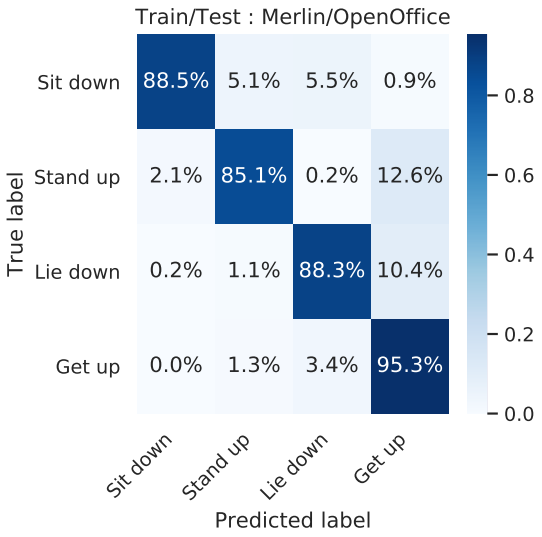


Figure 53: On the left: Prediction accuracy results for NN trained on Espoo Merlin and Shanghai datasets and validated on the Sunnyvale dataset. Lying down on a bed or floor mat is heavily confused with sitting down. The situation is common for a network trained and tested on different people and absent in the below confusion matrices for networks trained and tested on OpenOffice and Merlin datasets (Table 1) mostly collected from the same test subject.





5 Conclusion and discussion

5.1 Summary

Simple robotic devices to facilitate multi-environment channel state information (CSI) collection named *physical environment augmentation devices* (PEADs) have been developed and constructed.

In addition to prior foreign CSI data collection spaces, two contrasting spaces for activities data collection have been created locally. An activity classification dataset has been collected in each of the new expanses.

Both activity classification and human body localization are aimed to be performed by a final product. Since it is believed that the localization neural network has to be re-trained for every particular environment, it has been re-worked from ~58.8 to ~0.4 million parameters.

For the human activities classification part, a new sampling and preprocessing pipeline has been built. After Fourier spectrum amplitude-only representation had been experimented on, the CSI phase layer has been incorporated in samples.

A machine learning pipeline featuring on-the-fly configurable test/train data split and data augmentation has been created. Liberating activity samples from environment-specific information via data augmentation has proven to be more impactful compared to invented agnostic features of new activity classification neural network. On the other hand, applying cross-subcarriers and cross-links delta loss has seemingly prolonged the pre-plateau validation accuracy rise.

The performed datasets collection allowed cross-environment and/or cross-human testing via novel pipeline. This validated the environment-agnostic and lead to the discovery of human-dependent activity classification. Therefore, a necessity for human-diverse training data has been identified.

5.2 Ethical concerns

The large-scale adoption of Wi-Fi CSI sensing technology carries a risk of sovereignty transfer from private individual to an organization possessing means to process and benefit from volumes of collected data.

5.3 Further development

5.3.1 Human body localization

Further steps of NN size reduction during the training stage could include changing the structure of dense layers. For instance, a TreeConnect approach [Richter and Wattenhofer, 2018] replaces a shallow fully-connected layer with a deeper graph that processes input neurons in groups. The graph structure ensures that every output neuron is connected to every input neuron while decreasing asymptotic complexity from $O(n^2)$ to $O(n^{1.5})$.

In the current localization architecture (Figure 35), fully-connected layers are comparable in number of parameters with convolutional layers. In order to slightly

decrease the latter, spatially separable convolution id est decomposing one rectangular kernel into two one-dimensional perpendicular kernels may be used [Mamalet and Garcia, 2012]. The effect is more pronounced with larger original kernels. For instance, a 3×3 kernel with 9 parameters and 9 multiplications per point is decomposed to 3×1 and 1×3 kernels each with 3 parameters and multiplications per point yielding 6/9 or 2/3 improvement. At the same time, 5×5 kernel decomposition achieves 10/25 or 60% decrease.

It is, however, wise to keep in mind that many methods are double-edged swords and may carry various drawbacks. For instance, TreeConnect's sparsity trades-off better generalization ability with longer training time. For spatially separated orthogonal kernels it is inherently hard to detect diagonal features. All methods require hyperparameter tuning and some engineering effort to run properly. At the same time, embedded NN acceleration hardware may lack certain functionality, such as efficient point-wise or depth-wise convolution. Therefore, whether further NN optimization is necessary remains to be decided.

Other methods to reduce neural network size, including post-training ones are summarized by Deng et al. [2020].

5.3.2 Environment agnostic activity classification

As Section 4.4.7 demonstrates that the main differences in classification results originate from variations in test subjects.

Section 4.4.7 shows that although the network became environment agnostic, its classification results are still biased towards the person used for data collection. The proposed methods to make network human-agnostic are listed in Section 4.4.7. They boil down to diverse training data collection/generation. The latter could be approached with recording CSI samples maximally different test subjects, making a spectrogtram of their actions and interpolating across the differences.

Currently each subcarrier is equally normalized and fed as an input to NN. However, some of them have high amplitude and high variance, while other may have a problem with signal-to-noise ratio. For every bunch of subcarriers in CSI sample only the highest in amplitude/variance set may be selected for NN predicting. The rest could be used during the training phase as noise augmented data.

Current hyperparameters of the activity classification network are almost guaranteed to be sub-optimal due to limited intern-powered hyperparameters tuning. An automated hyperparameters search may find a noticeably better combination. In particular, the depth of the last dense block to fully-connected layers *transition bottleneck* may be increased from 4 channels to aim for greater representation capacity (Figure 4.4.5).

The experiments with FFT samples transformation may prove to be useful if (unlike in Section 4.4.3) it is performed on phase instead of amplitude and longer samples are taken into account. The effect of sample length and amplitude/phase input information for non-FFT samples can be seen from Figure 48.

Bibliography

John D. O'Sullivan, Graham R. Daniels, Terence M. P. Percival, Diethelm I. Ostry, and John F. Deane. Wireless LAN, November 1993. Patent US5487069A.

Xuefeng Liu, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. Wi-Sleep: Contactless sleep monitoring via WiFi signals. In *2014 IEEE Real-Time Systems Symposium*, pages 346–355, 2014.

Antonia M. Tulino, Angel Lozano, and Sergio Verdú. Impact of antenna correlation on the capacity of multiantenna channels. *IEEE Transactions on Information Theory*, 51(7):2491–2509, 2005.

Cecie Starr. *Biology: Concepts and Applications*. Core Biology. Brooks, 2005.

Thomas Rossing. *Springer Handbook of Acoustics*. Springer, 2014.

Kaoru Ashihara. Hearing thresholds for pure tones above 16kHz. *The Journal of the Acoustical Society of America*, 122, 2007.

Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. Capturing the human figure through a wall. *ACM Trans. Graph.*, 34(6), 2015.

Leonid A. Belov, Sergey M. Smolskiy, and Viktor Neofidovich Kochemasov. *Handbook of RF, Microwave, and Millimeter-wave Components*. Artech House, 2012.

Abdullah Khalili, Abdel-Hamid Soliman, Md Asaduzzaman, and Alison Griffiths. Wi-fi sensing: applications and challenges. *The Journal of Engineering*, 2020(3): 87–97, March 2020.

Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53, 2011.

Zheng Yang, Zimu Zhou, and Yunhao Liu. From RSSI to CSI: Indoor localization via channel response. *ACM Comput. Surv.*, 46(2), 2013.

Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM 2000*, 2000.

Fadel Adib and Dina Katabi. See through walls with WiFi! In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, page 75–86, 2013.

Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, page 27–38, 2013.

Yongsen Ma, Gang Zhou, and Shuangquan Wang. WiFi sensing with channel state information: A survey. *ACM Computing Surveys (CSUR)*, 52:1 – 36, 2019.

Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. Recognizing keystrokes using WiFi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1175–1190, 2017.

Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. Design and implementation of a CSI-based ubiquitous smoking detection system. *IEEE/ACM Trans. Netw.*, 25(6):3781–3793, December 2017.

Fusang Zhang, Daqing Zhang, Jie Xiong, Hao Wang, Kai Niu, Beihong Jin, and Yuxiang Wang. From Fresnel diffraction model to fine-grained human respiration sensing with commodity Wi-Fi devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), 2018a.

Pei Wang, Bin Guo, Tong Xin, Zhu Wang, and Zhiwen Yu. Tinysense: Multi-user respiration detection using Wi-Fi CSI signals. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, 2017a.

Xuyu Wang, Chao Yang, and Shiwen Mao. Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity WiFi. *ACM Trans. Intell. Syst. Technol.*, 9(1), 2017b.

Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2, 2019.

Linlin Guo, Lei Wang, Chuang Lin, Jialin Liu, Bingxian Lu, Jian Fang, Zhonghao Liu, Zeyang Shan, Jingwen Yang, and Silu Guo. Wiar: A public dataset for Wifi-based activity recognition. *IEEE Access*, 7:154935–154945, 2019.

David K. Cheng. *Fundamentals of Engineering Electromagnetics*. Dorling Kindersley India, 2014.

Sepp Hochreiter and Jürgen Shmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision RNNs for language recognition. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 740–745, 2018.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics, 2014.

Alexander Andreopoulos and John K. Tsotsos. 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117:827–891, 2013.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 550–558, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, 1999.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems 29*, pages 4898–4906. Curran Associates, Inc., 2016.

Clifford W. Ashley. *The Ashley Book of Knots*. Doubleday, Doran and Co, 1944.

Geoff Wilson. *Encyclopedia of Fishing Knots and Rigs*. Australian Fishing Network, 2003.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org, accessed: January 6th 2020.

David K. Cheng. *Field and Wave Electromagnetics*. Dorling Kindersley India, 2 edition, 2018.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bastiaan Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28:2222–2232, 2017.

Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices. *PACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 1, 01 2018.

Nicolas Vasilache, Jeff Johnson, Michaël Mathieu, Soumith Chintala, Serkan Piantino, and Yann LeCun. Fast convolutional nets with fbFFT: A GPU performance evaluation. *CoRR*, abs/1412.7580, 2015.

Andrew Lavin and Scott Gray. Fast algorithms for convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4013–4021, 2016.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *ArXiv*, abs/1602.07360, 2016.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018b.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

G.S. Daniels. *The Average Man?* Technical note WCRD. Wright-Patterson Air Force Base, 1952.

J. L. Flanagan and R. M. Golden. Phase vocoder. *The Bell System Technical Journal*, 45(9):1493–1509, 1966.

O. Richter and R. Wattenhofer. Treeconnect: A sparse alternative to fully connected layers. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 924–931, 2018.

Franck Mamalet and Christophe Garcia. Simplifying ConvNets for fast learning. In *Artificial Neural Networks and Machine Learning – ICANN 2012*, pages 58–65, 2012.

By Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020.

A Raw H matrix of a CSI data sample

```
array([[[[  27. -37.j,  103. -29.j,  -87. +12.j,   24.  -3.j],
         [ -57. -48.j,    0. +75.j,  130. -10.j,   24. -86.j],
         [  71. +46.j,    1. +68.j, -128. +52.j,   25.  +3.j],
         [  66. +96.j,  -39. +10.j,  -56. +51.j,  -20. +20.j]],

        [[  64. -10.j,  137. +69.j, -104. -80.j,   33. +25.j],
         [ -20.-110.j,  -72. +81.j,  155.+124.j,  114. -68.j],
         [  25.+118.j,  -68. +70.j, -186. -77.j,   15. +27.j],
         [ -26.+166.j,  -54. -28.j, -110.  -1.j,  -49.  +2.j]],

        [[  76. +50.j,   74.+183.j,  -32.-169.j,    9. +59.j],
         [  73.-117.j, -142. +23.j,   44.+257.j,  166. +28.j],
         [ -83.+144.j, -130.  +9.j, -112.-252.j,   -6. +42.j],
         [-173.+139.j,  -33. -77.j, -114.-109.j,  -47. -52.j]],

        ...,

        [[  83.  +1.j,  175.+115.j,  -15.-118.j,  -36. +57.j],
         [ 131. -70.j, -172.  -2.j,   -4.+121.j,  115. +63.j],
         [ -61.-161.j,   75. -69.j,   89. +52.j,  -50. +17.j],
         [-231. -12.j,   81. -93.j,  -18. -92.j,  -17. -18.j]],

        [[  48. +40.j,   43.+152.j,   43. -68.j,  -45. +15.j],
         [ 102. +24.j,  -94. -85.j,  -62. +69.j,   31. +95.j],
         [  47.-117.j,   83.  +2.j,   29. +79.j,  -23. -18.j],
         [-121.-122.j,   91. -12.j,   37. -60.j,    9. -24.j]],

        [[  10. +41.j,  -44. +96.j,   51. -19.j,  -26. -15.j],
         [  35. +63.j,  -10. -86.j,  -68. +11.j,  -26. +61.j],
         [  77. -38.j,   42. +39.j,  -23. +52.j,  -12. -27.j],
         [  -7.-116.j,   52. +35.j,   45. -16.j,    8.  -9.j]]],

        ...,

H-matrix example shape: (27198, 104, 4, 4), where:
shape dimension 1 - number of samples in the recording
shape dimension 2 - number of sub-carriers
(4, 4) - dimensionality of MIMO paths matrix
         with 4 AP antennas and 4 spatial streams
```

B Two sets of audio instructions for a test subject to follow

timestamp: begin	timestamp: end	duration [ms]	total [s]	label
1564015509601	1564015524601	15000	15	UNDEFINED
1564015524601	1564015527601	3000	18	STAND_TO_SIT
1564015527601	1564015528601	1000	19	UNDEFINED
1564015528601	1564015537601	9000	28	SIT
1564015537601	1564015538601	1000	29	UNDEFINED
1564015538601	1564015541601	3000	32	SIT_TO_STAND
1564015541601	1564015542601	1000	33	UNDEFINED
1564015542601	1564015551601	9000	42	STAND
1564015551601	1564015552601	1000	43	UNDEFINED
1564015552601	1564015555601	3000	46	STAND_TO_SIT
1564015555601	1564015556601	1000	47	UNDEFINED
1564015556601	1564015565601	9000	56	SIT
1564015565601	1564015566601	1000	57	UNDEFINED
1564015566601	1564015569601	3000	60	SIT_TO_STAND
1564015569601	1564015570601	1000	61	UNDEFINED
1564015570601	1564015579601	9000	70	STAND
1564015579601	1564015599601	20000	90	UNDEFINED
1564015599601	1564015602601	3000	93	STAND_TO_SIT
1564015602601	1564015603601	1000	94	UNDEFINED
1564015603601	1564015612601	9000	103	SIT
1564015612601	1564015613601	1000	104	UNDEFINED
1564015613601	1564015616601	3000	107	SIT_TO_STAND
1564015616601	1564015617601	1000	108	UNDEFINED
1564015617601	1564015626601	9000	117	STAND
1564015626601	1564015627601	1000	118	UNDEFINED
1564015627601	1564015630601	3000	121	STAND_TO_SIT
1564015630601	1564015631601	1000	122	UNDEFINED
1564015631601	1564015640601	9000	131	SIT
1564015640601	1564015641601	1000	132	UNDEFINED
1564015641601	1564015644601	3000	135	SIT_TO_STAND
1564015644601	1564015645601	1000	136	UNDEFINED
1564015645601	1564015654601	9000	145	STAND
1564015654601	1564015660601	6000	151	UNDEFINED

timestamp: begin	timestamp: end	duration [ms]	total [s]	label
1549178395068	1549178410068	15000	15	UNDEFINED
1549178410068	1549178600068	190000	205	WALK
1549178600068	1549178605068	5000	210	UNDEFINED

C Delays between CSI reports in a whole standard recording

Below are subplots with delays between individual CSI reports received within the 155.9 seconds recording. This recording has not yet been sliced into single activity CSI samples. It includes all physical activities performed according to a whole audio instruction. An example of such audio instruction can be found in Appendix B

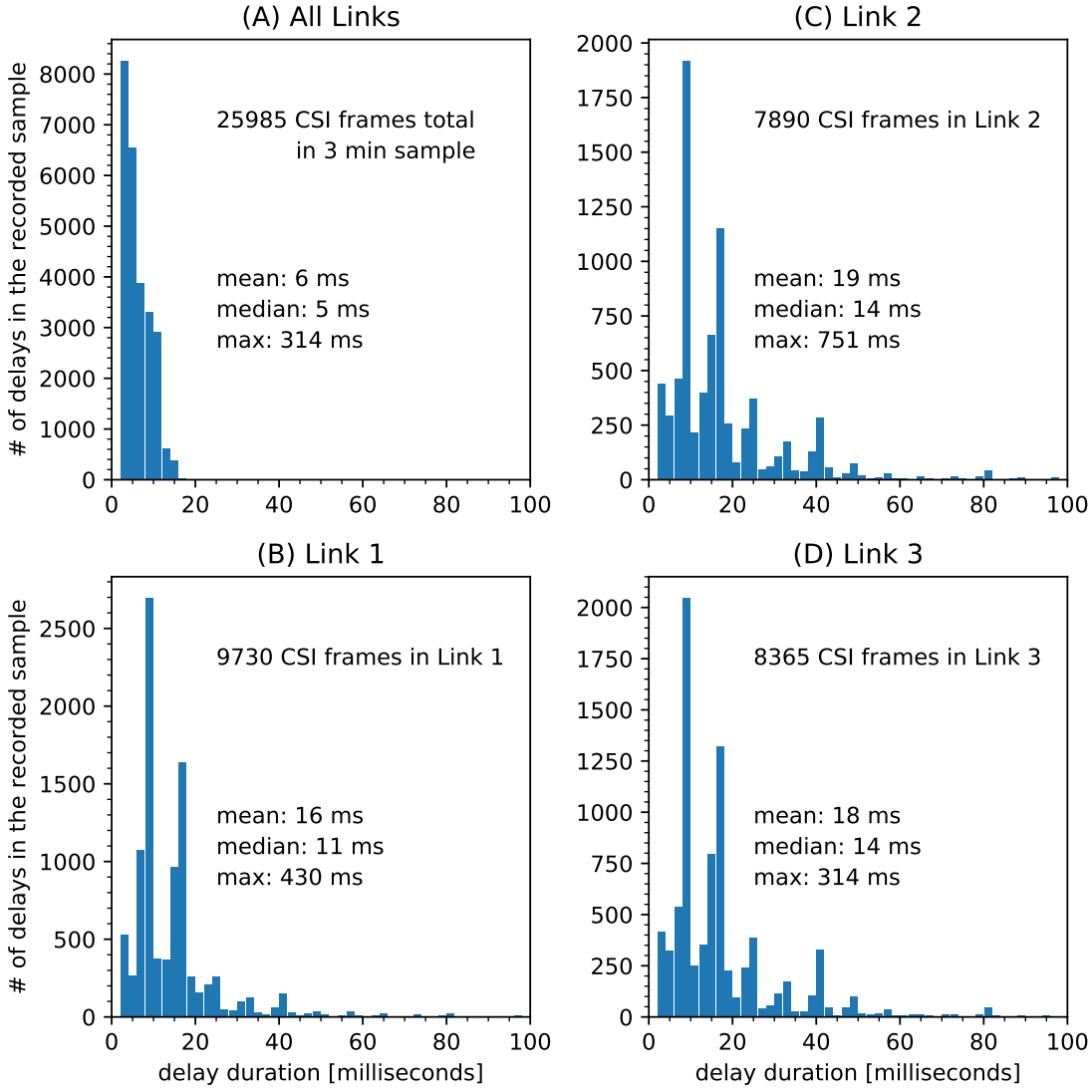


Figure C1: (A): Delays between reports from all 3 AP-STA links, zipped in single stream. (C-D): Delays between individual AP-STA link reports. The horizontal [ms] scale is shared between all subplots.